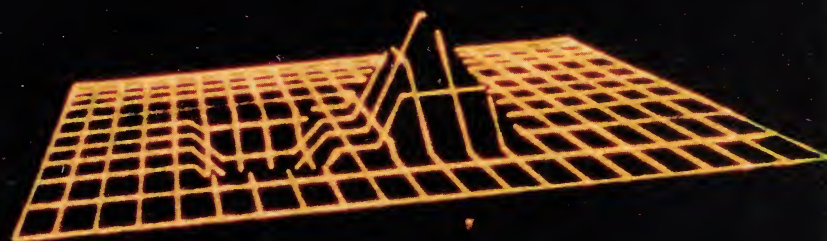
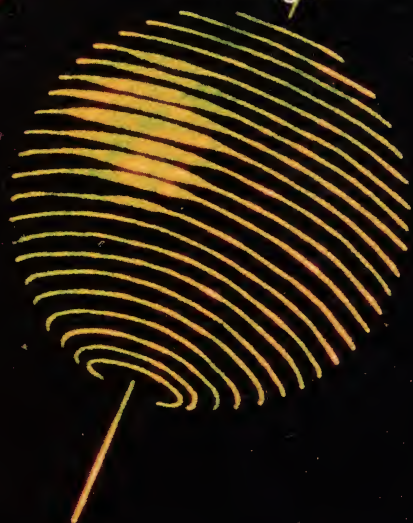


Vol. 1, Issue 9  
Dec/Jan'85

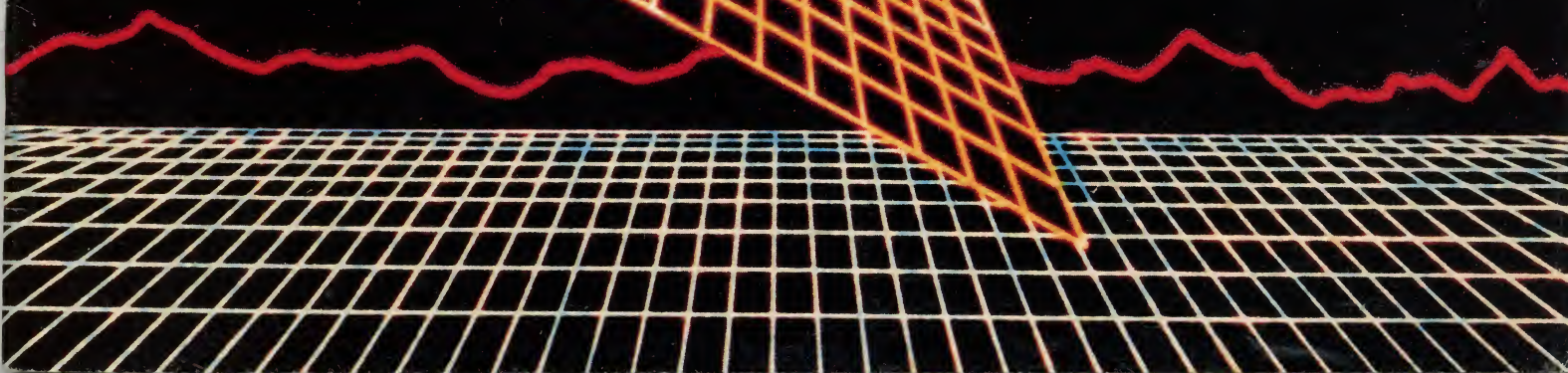
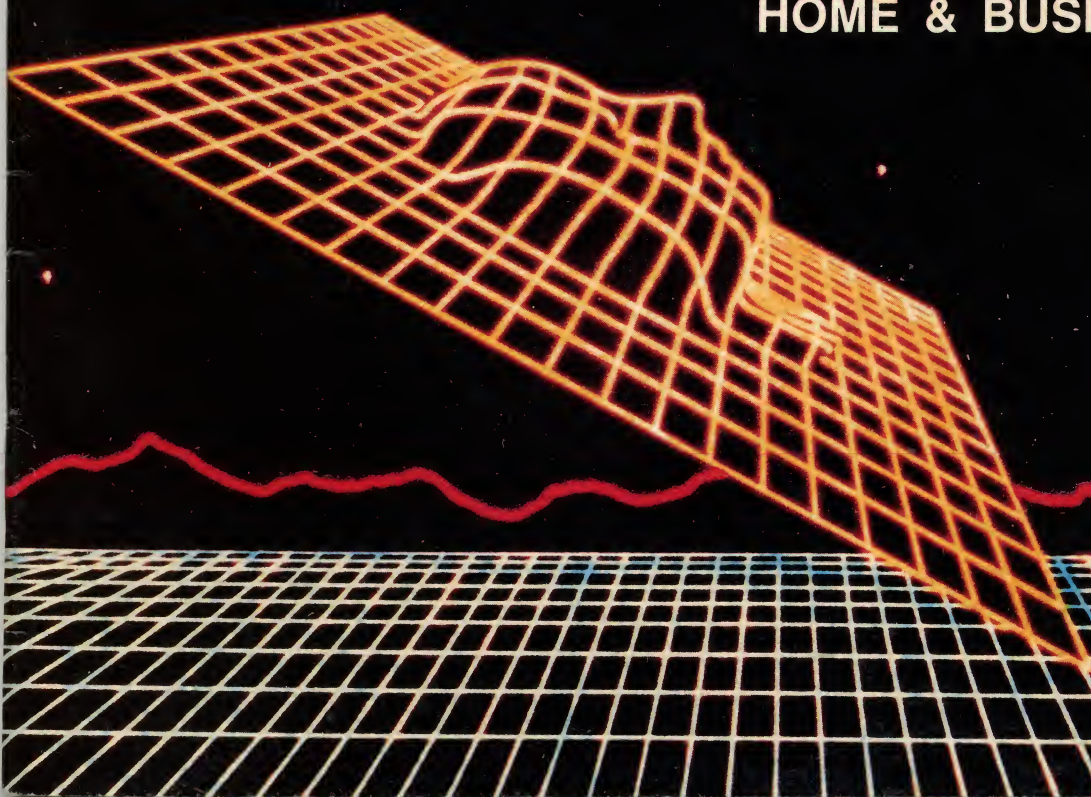
\$2.50 U.S.  
\$3.25 Cdn.

# ROOM

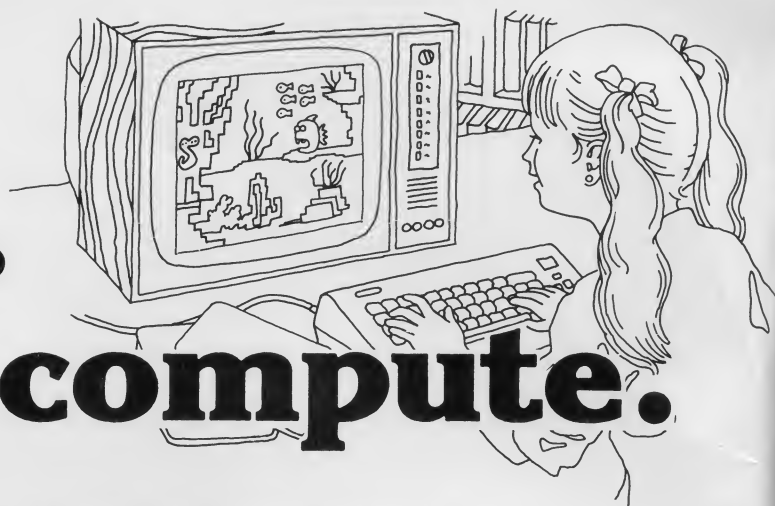
The Magazine That Brings The ATARI™ Computer to Life!



**ATARI'S TIMING SYSTEM  
HOME & BUSINESS PROGRAMS  
LINES OF ACTION  
AIR ATTACK**



# See Jane. See Jane compute.

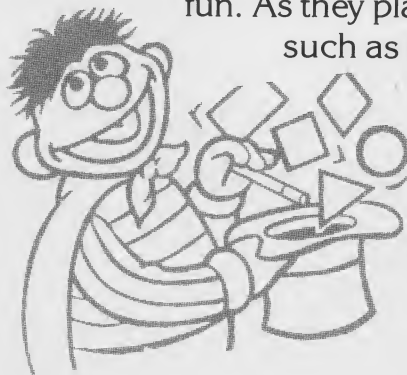


These days, kids like Jane -- kids like your own -- are playing with computers. That's why we've stocked up on early education programs from CBS Software.

You see, CBS Software can inspire kids to be their very best. How?

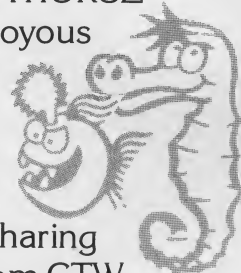
For one thing, many of the programs were developed by Children's Television Workshop, the people who created *SESAME STREET*™. Others are the brain-children of education and entertainment specialists who share the idea that learning can and should be fun. As they play with programs

such as *ERNIE'S MAGIC SHAPES*™\* and *BIG BIRD'S SPECIAL DELIVERY*™\*, children interact with *SESAME STREET*™



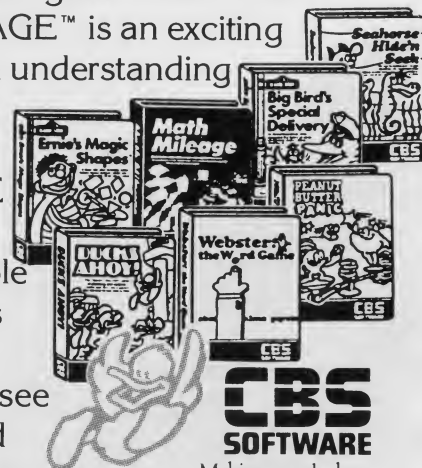
characters they know and love while learning about shapes, colors... and the computer.

*DUCKS AHOY*™ and *SEA HORSE HIDE'N SEEK*™ are gentle, joyous activities that help kids learn timing, anticipation, color and spatial relationships. *PEANUT BUTTER PANIC*™ encourages teamwork and sharing in another dynamic game from CTW.



*MATH MILEAGE*™ is an exciting "race" toward an understanding of basic math concepts. *WEBSTER: THE WORD GAME*™ is fun for the whole family as it hones spelling skills.

Come in and see why we're excited about the programs from CBS Software. And bring along your Jane. Or David. Or Amy. Or...



**CBS  
SOFTWARE**  
Making you the best.

Available for most popular home computers.

## DISTRIBUTED BY:



**MICROWEST**

**DISTRIBUTING INC.**

1916 Ellis St., Bellingham, WA., 98225  
(206)671-1600

106 Donaghy Ave., N. Vancouver, B.C., V7P 2L5  
(604)984-9191 (Order Desk (800)663-5993)





**Vol. 1, Issue 9**  
**DEC/JAN '85**  
**ROM**  
**STAFF**

**EDITOR**  
Peter Ellison

**Publishers**  
Peter Ellison  
Bob Cockroft

**Art Director**  
Dave Bacon

**Technical Division**  
Bob Cockroft  
Jack Chung

**Photographer**  
Jason Cockroft

**Contributors:**  
Kevin Greggain  
Sol Guber  
Paul Knapp  
Stephen Everett  
Gavin Bamber  
Joel Ferguson

**ROM Magazine Corporation is in no way affiliated with Atari. Atari is a trademark of Atari, Inc.**

## FEATURE/ARTICLES

Interview: Chuckles (Charles Bueche) .....	Peter Ellison	14
Atari's Timing System .....	Bob Cockroft	25
Examining Atari DOS .....	Bob Cockroft	44
Synapse: A Business Approach .....	Peter Ellison	23
Lister Plus .....	Peter Ellison	38
Sparta DOS and US 1050 Doubler .....	Peter Ellison	58

## PROGRAMS

Air Attack .....	Bob Cockroft	16
Lines of Action .....	Sol Guber	32
Home and Business Programs .....	Stephen Everett	39

## COLUMNS

Editorial .....	Peter Ellison	4
Letters .....	Peter Ellison	4
Jake the Software Dude .....	Jason Cockroft	59
Strategy Zone .....	Bob Cockroft	6
Yellow Brick Road .....	Peter Ellison	12
Books On The Shelf .....	Peter Ellison	31
Game Reviews .....		54
Action! Corner .....	Kevin Greggain	8

Domestic Subscriptions: 1 yr.(6 issues) bimonthly \$12 U.S.; 12 issues \$23 U.S. 1 yr. Disk & Magazine \$60 U.S. Send Subscription orders to: ROM Magazine, P.O. BOX 252, Maple Ridge, B.C. V2X 7G1. CANADA. (604) 462-9128 or 462-9177.

Nothing in this magazine may be reproduced without the permission of the publisher. Program listings should be in printed form. Articles should be as a type copy in upper and lower case with double spacing. By submitting articles to ROM articles accepted for publication will become the exclusive property of ROM. If an article is not accepted, the only way it will be returned is; if a self-addressed envelope is enclosed. PRINTED IN CANADA. Second Class mail registration pending.

# EDITORIAL

Winter is upon us once again, and our interests begin to turn from outdoor activities to warmer indoor hobbies, such as using computers. With Jack Tramiel dropping the prices of the Atari computer and disk drives, more people will be turning to the Atari; consequently, there will be a larger number of users for User Groups.

This brings me to the point of User Groups. In this issue we have sent out a number of Questionnaires to get user insight on the Atari. We, at ROM, want to be 'the magazine for Atari Computer Users' (User Groups). Starting with this issue, we are allocating a page to User Groups so they can make announcements of activities, or make themselves known for User Group Communication. This column will be called "The User Group Network," so, if you have any comments or suggestions regarding it, send them in care of this column.

We have an excellent new filing system, (Reviewed in this issue), called, "Synfile +," in which we will keep track of all our subscribers and standing orders.

Each subscriber gets a magazine with a label, on the outside of his or her envelope. On the bottom of each label, starting with this issue, will be a number designating the issue with which the person's subscription runs out. For example, if, at the bottom of your label it reads 'D13,' it means your subscription is a disk version subscription, and it runs out with the thirteenth issue. If there is no 'D' before it, it means that you are not a disk subscriber. This arrangement will help subscribers avoid missing out on following issues.

This issue contains an assembly language program called, 'Air Attack,' a two-player, air-sea battle. This game, written by Bob Cockroft, should keep you entertained for hours to come. Also, in this issue, is an interview with Chuckles (Origin Systems), "Lines of Action" by Sol Guber, and a Tutorial on Atari's timing system. In this issue we have also started a new column called, 'The ACTION! Corner.' This section will contain programs and tutorials that are written in ACTION!. We feel this language is the one a person should buy if he is fed up with the speed of BASIC, and doesn't want to be bothered learning assembly language. One last thing before closing: Starting with this issue, we are putting dates on our magazine. With these improvements and those yet to come, I hope we can make ROM your number one Atari magazine.

Peter Ellison  
Editor/Publisher

# LETTERS

Dear ROM:

Recently while visiting a computer store, I saw copies of ROM magazine. After a quick glance, I was very impressed and managed to obtain a copy of each of the first eight issues.

One of the things which impresses me about ROM is that a good effort is made to explain the program code, and not just to print the program without any explanation, other than how to use the joysticks at run-time! The series on 'Display Lists' is a good example. The one criticism I have is the heavy game emphasis, but that is probably a strong point with most readers.

Norman J. Hill  
St. Marys, Ontario

Dear Norman:

Thankyou for your kind words, and we'll try to continue in giving you good tutorials. We hope that our emphasis, in time, will be more on tutorials and teachings new things to Atari Users.

Dear ROM:

I want to thank you for your prompt reply to my inquiries, and with that kind of service I am sure your magazine will be around for a long time.

Michael Cote  
Winnipeg, Manitoba

Dear Michael

Thankyou for your encouraging letter. We will be around for a long time, you can count on that!

Dear ROM:

I just received my first issue of ROM magazine. All I can say is GREAT JOB! I'm very impressed to say the least.

Harold Higgins  
Pocatello, Idaho

Dear Harold:

On behalf of the ROM staff, I thank you. Each issue we try to improve our quality over the one before. We feel with this policy we can't lose.



## LOTSABYTES DECLARES WAR!

**WAR** on high prices! We're going to put an end to the software price 'ripoff'. And **YOU** can help! Just keep those orders coming while you continue to enjoy the **quality, quantity, selection and low prices** that you deserve. Our National Public Domain Copy Service will save you time, tedious work, and money. And our **exclusive** distribution of **sharply discounted** commercial programs will bring you some of the finest programs for the lowest possible price, usually 50% off retail! You continue to get **FREE BONUSES** with each purchase of three or more disks.

### PUBLIC DOMAIN SOFTWARE

<b>#1 GAMES</b> Two full disk sides packed with over 25 games including some Arcade quality. \$7.95	<b>#2 UTILITIES</b> 25 powerful programs to help you get the most out of your Atari computer. \$7.95	<b>#3 AMS MUSIC</b> 25 Advanced Musicsystem files including a new Player program. 2 sides. \$7.95	<b>#4 GAMES</b> All different! 14 more better games on 2 disk sides. Some Arcade types. \$7.95	<b>#5 EDUCATION</b> Loaded with 28 programs on 2 disk sides. Fun learning for the whole family. \$7.95
<b>#6 AMS MUSIC</b> 25 all-time favorites with a Player program. Two sides. \$7.95	<b>#7 GAMES</b> Two disk sides packed with 14 more great games. Some Arcade types. \$7.95	<b>#8 UTILITIES</b> 17 more power-packed utilities to help unleash the full potential of your Atari. \$7.95	<b>#9 GAMES</b> <b>NEW!</b> Two full sides filled with some of the best and most recent. Some Arcade. \$7.95	<b>#10 UTILITIES</b> <b>NEW!</b> A new assortment of great and powerful programs. Don't miss it! \$7.95

### LotsaBytes EXCLUSIVES

#### ADVANCED MUSICSYSTEM II

by LEE ACTOR  
Allows you to create music with your Atari computer! All new machine code.

- \* Control over pitch duration, envelope, dynamic level, meter, tempo and key.
- \* 4 independent voices
- \* 5½ octaves per voice
- \* Save up to 8200 notes
- \* Custom DOS
- \* FULL instructions
- \* 24K disk

Originally \$29.95

Only \$14.95

#### ORIGINAL ADVENTURE

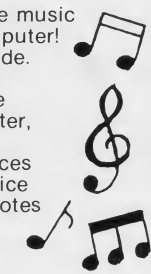
by Bob Howell  
For all Atari computers. The Original Colossal Cave Adventure faithfully reproduced from the 'main-frames'.

This is the one that launched the whole Adventure craze of today!

- \* Two mazes
- \* 130 rooms
- \* Deadly Dragons
- \* Nasty Dwarves
- \* Tenacious Troll
- \* The Pirate & More!
- \* 86 coded hints
- \* SAVE/RESUME
- \* 40K disk or 32K tape

Originally \$29.95

Only \$14.95



#### QUALITY WORD PROCESSING

**ESI WRITER!** At last a brand-new Word Processor that has more features and is easier to use than anything else available for the Atari. Easy for the beginner to use, it asks questions and remembers the answers. ESI WRITER is so sophisticated that it has many more features we don't even have room to mention! Works with ANY Atari.

- \* Reads any text file
- \* Built in Help screen
- \* Very fast!
- \* Works with ANY printer
- \* Instant top, bottom or text location without scrolling!
- \* Every printer feature
- \* DISK ONLY (Any Atari)
- \* Search and replace
- \* Block move text
- \* Page eject/start
- \* Set margins/lines etc.
- \* Full justification
- \* Print headers etc.
- \* Block delete etc.
- \* Change video color
- \* Over 50 pages of docs and tutorials

**TRUST US ON THIS ONE! YOU WILL LOVE IT!**

Originally \$49.95

LotsaBytes price \$24.95

#### \* \* FREE BONUSES \* \*

If you purchase any 3 or more disks at a time you may choose any 1 of the following disks **FREE!!**

- The Atari XL TRANSLATOR DISK that enables XL owners to use most 400/800 software. **FREE!!**
- An all different AMS MUSIC disk with Player. **FREE!!**
- Your choice of one of the P.D. disks -- #1, #2, #3, #4, #5, #6, #7, #8, #9, or #10 (specify one) **FREE!!**

Full 100% Replacement guarantee. Any disk found to be defective will be replaced free and we will also refund your return postage. All orders shipped by First Class U.S. Mail. Add \$1.95 shipping and handling for 1 to 5 disks. Add \$2.95 for 6 to 12 disks. California residents add 6% sales tax. Outside of U.S.A. and Canada add 15%. U.S. Funds only. We accept checks or Money Orders. Sorry, no COD or Charge Cards. Allow three weeks for personal checks to clear.

# LOTSABYTES

15445 Ventura Blvd., Suite 10  
Sherman Oaks, CA 91413

Atari is the registered trademark of Atari, Corp.

## GREAT GAMES!

### LotsaBytes Exclusives!

**SPACE GAMES:** Three games for **one** low price! In **Aliens** you can't get them all and the pace keeps getting faster. When you do get rid of most of them, you are left in a space quadrant peppered with mines. Will you **Survive?** If you do, you must penetrate the alien's spaceship, survive a **Robot Attack**, and get back your stolen 'cloaking' device! Interested?

\$24.95 list

LotsaBytes price: \$12.95

**THE BEAN MACHINE** by Steve Robinson is an Award Winning Arcade game that will drive you crazy balancing a series of beams while trying to get all the beans to roll down, without touching, all the while avoiding 'strange creatures' who drop in to steal the beans. It's addicting!

\$24.95 list

LotsaBytes price: \$12.95

**DIGGERBONK**, another Award Winning game by Steve Robinson, challenges you to find your way through a continuously scrolling maze while avoiding some really strange creatures. Along the way you will need to Bonk some of them, but watch out for the bombs.

\$24.95 list

LotsaBytes price: \$12.95

**GUESS WHAT'S COMING TO DINNER** lets you try to maneuver a snake through 7 levels if you can keep it from starving or being electrocuted. Lots of surprises! One or two players.

\$24.95 list

LotsaBytes price: \$12.95

\* \* \* \*

### CREATIVE LEARNING ADVENTURES

**Ages 4 to 10 — Disk only**

1. Hours of **educational** fun playing 3 exciting creative adventures with a friendly alien learning about our planet Earth. Hand/eye co-ordination, drawing, and music skills are emphasized.

\$24.95 list

LotsaBytes price: \$12.95

2. Four challenging learning games that are the favorites of our friendly alien. Helps your child to develop logical reasoning ability.

\$24.94 list

LotsaBytes price: \$12.95

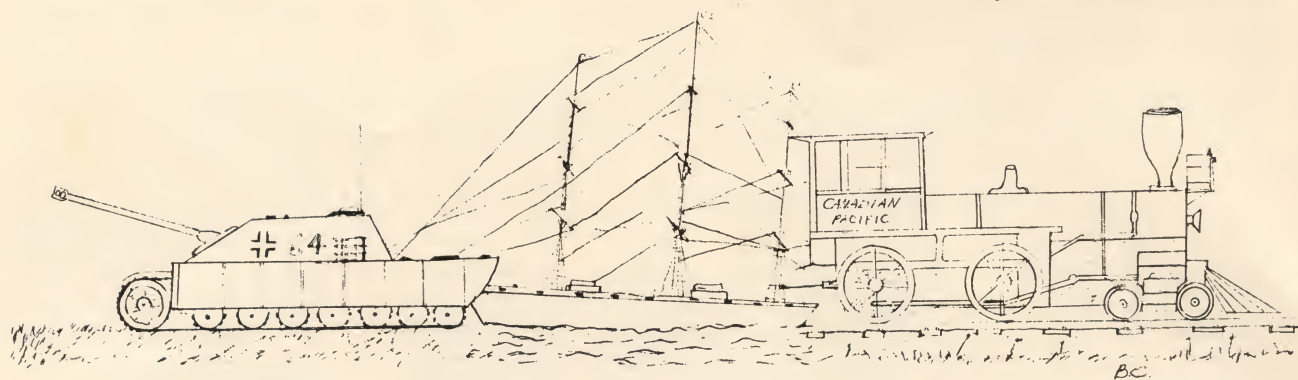
3. These 3 Fun-Day learning games will help with intellectual development, hand/eye co-ordination, logic, spatial, and analytical abilities.

\$24.95 list

LotsaBytes price: \$12.95

# Strategy Zone

by Bob Cockcroft



## Rome and the Barbarians

### KRENtek Software

'Rome and the Barbarians' by KRENtek Software is a game that recreates the period when Rome was in a struggle against eastern barbarian tribes. At this time, the empire was in decline, and the once powerful legions were losing their military superiority. As Emperor, it will be your job to resist these invaders, and preserve the empire.



The Roman and Barbarian forces are displayed on a large scrolling map of western Europe. The map, being the surface on which the game is played, scrolls at a speed which is both rapid and playable. Cities, mountains, rivers, and other geographical features are presented in a high degree of detail. The legions are given orders from a joystick controlled command box in a manner similar to that of 'Eastern Front'. As in a number of other wargames, troops can be ordered to move in a predetermined path while the player's attention is focused elsewhere. This technique has the obvious advantage of being time efficient. But what really makes the mechanics of movement different in this game, is that the player is able to combine units. One of the most frustrating aspects of some wargames

is the inability to concentrate strength at a particular point. Although a player may have many units at his disposal, the maximum power that can be allocated to any point is the combat strength of one unit. Because the game allows the combining of legions, it has gotten around this problem.

'Rome and the Barbarians' depicts this ancient struggle in 3 different ways: militarily, politically, and financially.

This conflict is fundamentally a military one. As a game of ancient warfare 'Rome and the Barbarians' does a fairly good job of entertaining and of creating an environment in which numerous strategic options exist: there seem to be few set patterns. The empire can be defended in a number of different ways, although the protection of Italy is essential. Unlike other wargames, the occupation of cities is far more important than the control of territory. 'Rome and the Barbarians' uses a real time combat system. There are not any turns; action is continuous. Because commands are given while the 'play' is in progress, the game moves quickly. The Emperor must continually move the control box around the map, supplying orders to all his active legions. Although the action may progress a little too quickly, 'Real time' gives the game a more realistic presentation.

One of the less attractive aspects of this game is that it uses a multi-screen system. Because one must refer to other displays for information, some of the continuity of the game is lost. The player should not be forced to divide his attention from the main map display on a regular basis.

On another level, 'Rome and the Barbarians' is an exercise in diplomacy. Some barbarian tribes can be paid to fight for the empire. However, Rome has a limited supply of funds that can be allocated for bribes. Therefore, it is crucial to correctly choose which barbarians are to be made allies. If carefully picked, paid barbarians can be made to do the majority of the fighting, while the Legions preserve their strength.

*Continued on Page 60*



# NAPOLEON AT WATERLOO



Relive this classic battle, as you lead the French against the combined strength of the British and Prussian armies. The crisp graphics and simple to use command system make this real-time game a joy to play. A must for anyone interested in Napoleonic Warfare.

## **NAPOLEON AT WATERLOO**

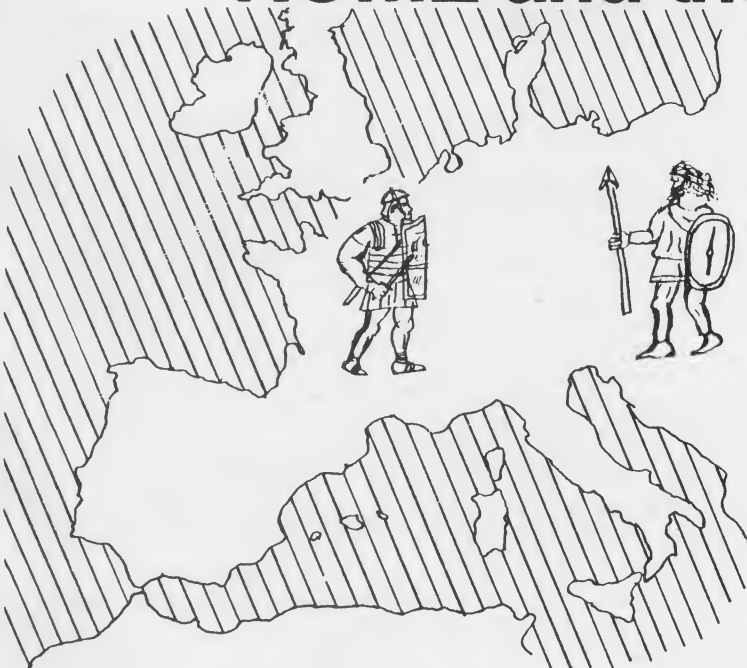
on 32K cassette and disk for ATARI computers \$34.95

(Kansas residents add \$1.75 sales tax)

# **KRENtek**

# **Software**

## ROME and the Barbarians



The year is 400 A.D. and only you, as supreme commander, can save the Roman Empire from the onslaught of the German tribes. This real-time, graphics oriented game looks like it belongs in an arcade. But, as a military/economic simulation, it will challenge your mind like no arcade game ever could.

## **ROME and the Barbarians**

on 32K cassette and disk for ATARI computers \$34.95

(Kansas residents add \$1.75 sales tax)

KRENtek Software

P.O. Box 3372

Kansas City, KS 66103

(913) 362-9267

ATARI is a registered trademark of ATARI, INC.





# ACTION! CORNER

by Kevin Greggain

Computer languages come and go with time because of many factors. One of these is speed. How fast a program will run is very important to someone who is writing a fast-action game. If it was written in BASIC, the game would be too slow. To take the big step and write the game in assembly language would take a long time with a lot of hard work. This is where ACTION! comes along. Although it is a high-level language, like BASIC, when compiled RUNs quickly like a low-level language.

The thing that makes this language so unique is how easy a BASIC programmer could switch over to using it. A complete BASIC program could be converted over to ACTION! and may run over 10 times as fast. In this article, I'm first giving you a program that will run a number of spectacular graphic demos. These demos, if written in BASIC, would be so slow they'd be boring, but in ACTION!, they are fast and colorful. Each demo has comments to make them easier to understand.

The second program is just a short one that moves the character set from ROM to RAM for use in your own custom character set. In the next issue a complete listing to a game written in ACTION! will be given.

```

MODULE                                ;WE'RE GOING TO D
ECLARE SOME                           ;GLOBAL VARIABLES
BYTE C4=704,                          ;C5=705,
FOR USE BY THE                        ;C6=706,
C5=705,                               ;C7=707,
C6=706,                               ;C8=708,
REGISTERS FOR MODE 10                ;C9=709,
C7=707,                               ;C10=710,
C8=708,                               ;C11=711,
C9=709,                               ;C12=712,
C10=710,                              ;LM,RM,
C11=711,                              ;BM,RO,
C12=712,                              ;C IS THE POINTER
LM,RM,                                ;VARIABLE USED FOR THE COLOR
BM,RO,                                ;ATT=77,
LOOP,C=[1],                          ;ATTRACT MODE AT
VARIABLE USED FOR THE COLOR          ;77 IS ALWAYS RESET TO 0
ATT=77,                               ;LOOP1,LOOP2,
77 IS ALWAYS RESET TO 0              ;CONSOLE=53279 ;THE CONSOLE KEYS
LOOP1,LOOP2,                         ;START SELECT OPTION
CONSOLE=53279 ;THE CONSOLE KEYS
START SELECT OPTION

PROC COL()                            ;THIS ROUTINE SET
S THE SEQUENTIAL BUT RANDOM COLORS TO
DRAW WITH                             ;DRAW WITH
BYTE NUM

NUM=Rand(14)*14

C4=0 C5=NUM C6=NUM+2 C7=NUM+4 ;SETS C
OLORS TO COLOR BASE TO COLOR BASE+14
C8=NUM+6 C9=NUM+8 C10=NUM+10 ;SAME A
S BASIC POKE
C11=NUM+12 C12=NUM+14

RETURN

PROC SCROLL()                        ;THIS I
S THE ROUTINE TO PUT IT ALL TOGETHER O
NCE IT'S DRAWN
BYTE WSYNC=54282,                   ;WAIT S
YNC LOCATION                         ;VNC LOCATION
VCOUNT=54283,                       ;VERTIC
AL LINE COUNTER

```

```

CLOCK=20,                            ;RTC CL
OCK AT 18,19,20 USED FOR TIMING     ;
CLOCK2=19,                           ;JIFFY
CLOCK3=540,                           ;
COUNTER                              ;SCREEN
C01=$D012,                           ;
REGISTER FOR GRAPHICS 10             ;TEMPOR
TEMP                                  ;ARY STORAGE VARIABLE FOR LOCAL USE
CLOCK2=0 CLOCK3=240                  ;SET CL
OCK TIMERS                           ;
DO                                    ;BEGIN
LOOP                                ;COMPAR
IF CLOCK3=0 THEN                     ;E 540 WITH 0 (CLOCK COUNTS BACKWARDS F
ROM 240 AS ORIGINALLY SET)
TEMP=C12                             ;STORE
REGISTER INTO TEMP..SIMILAR TO PHA IN
ASSEMBLY)                            ;THESE
C12=C11                              ;FOLLOWING MOVES SWAP THE COLOR REGISTE
RS
C11=C10
C10=C9
C9=C8
C8=C7
C7=C6
C6=C5
C5=TEMP
CLOCK3=2                             ;RESET
540 TO 2 (2 60ths OF A SECOND)
FI
IF VCOUNT<=61 THEN                  ;CHECK
54283 FOR <=61 (APP MID SCREEN)     ;STORE
WSYNC=1                              ;
AT WSYNC                             ;STORE
C01=VCOUNT+CLOCK                    ;COLOR IN REGISTER $D012
COLOR IN REGISTER $D012              ;IS CON
DITION TRUE                          ;NO SO
GO TO SECOND PART                    ;BIGGER
ELSEIF VCOUNT>=62 THEN              ;THAN MID SCREEN?
WSYNC=1                              ;STORE
AGAIN                                ;STORE
C01=VCOUNT-CLOCK                    ;COLOR -CLOCK FOR REVERSE SCROLLING EFF
ECT                                  ;END OF
FI                                    ;CHECK
UNTIL CLOCK2=10 OR CONSOLE=6        ;LOOK F
OR CLOCK 19=10 OR START KEY PRESSED (C
ONSOLE=6 =START)
DO                                    ;END OF
SCROLL ROUTINE                      ;TURN O
ATT=0                                ;FF ATTRACT IF ON
RETURN                               ;ATTRAC
T ENABLES AFTER LOCATION 77>=128

PROC DR1()                           ;THE DR
# PROCEDURES ARE USED BY THE MAIN ROUT
INE TO DRAW THE IMAGES.
LM=0 RM=79 BM=191 RO=0              ;WE SET
THE LEFT RIGHT BOTTOM AND TOP MARGINS
FOR USE WITHIN THE PROGRAM

GRAPHICS(10) C4=0                   ;SET UP
MODE FOR SCREEN AND SET SCREEN COLOR
TO 0 (BLACK)

COL()                                ;CALLS
BACK TO THE COL() PROCEDURE WHICH SETS
THE COLOR REGISTERS

DO
COLOR=C
PLOT (LM,RO)
DRAWTO(RM,RO)
DRAWTO(RM,BM)

```

```

DRAWTO(LM,BM)
DRAWTO(LM,RO)

C==+1 IF C>8 THEN C=1 ;THIS C
HECKS FOR C AND RESETS IF COLOR > 8 TO
1
FI

LM==+1 RM==--1 BM==--1 RO==+1

UNTIL LM=79 OD

RETURN

PROC DR2()

LM=0 RM=79 BM=191 RO=0
GRAPHICS(10) COL()

DO

FOR LOOP=LM TO RM DO
COLOR=C
PLOT(LOOP,RO) C==+1
IF C>9 THEN C=1 FI
OD

FOR LOOP=RO TO BM DO
COLOR=C
PLOT(RM,LOOP) C==--1
IF C<1 THEN C=9 FI
OD

LOOP=RM

DO
COLOR=C
PLOT(LOOP,BM) C=C+1
IF C>9 THEN C=1 FI
LOOP==--1
UNTIL LOOP=LM
OD

LOOP=BM

DO
COLOR=C
PLOT(LM,LOOP) C=C-1
IF C<1 THEN C=9 FI
LOOP==--1
UNTIL LOOP=RO
OD

LM==+1 RM==--1 BM==--1 RO==+1
UNTIL LM=40 OD
LM=39 RO=95 BM=95 RM=39
DO
COLOR=C
PLOT(LM,RO) DRAWTO(RM,RO)
DRAWTO(RM,BM) DRAWTO(LM,BM)
DRAWTO(LM,RO)
C==--1 IF C<1 THEN C=8 FI

LM==--1 RM==+1 RO==--1 BM==+1

UNTIL LM=10 OD

RETURN

PROC DR3()

BYTE X,Y

GRAPHICS(10)
COL()
FOR LOOP=0 TO 100 DO

COLOR=RAND(9)+1
X=RAND(79)
Y=RAND(100)

PLOT(X,Y)
OD

LM=0 RM=79 BM=191 RO=10

DO

COLOR=C
PLOT(LM,RO) DRAWTO(LM,BM)
DRAWTO(RM,BM) DRAWTO(RM,RO)
C==--1 IF C<1 THEN C=8 FI

```

```

LM==+1 RM==--1 BM==--1 RO==+3

UNTIL LM=37 OD

RETURN

PROC DR4()

GRAPHICS(10) COL()

FOR LOOP1=30 TO 50 STEP 2

DO
COLOR=C
C==+1 IF C>8 THEN C=1 FI

FOR LOOP2=LOOP1 TO LOOP1+2
DO
PLOT(5,LOOP2) DRAWTO(74,LOO
P2)
OD
OD

FOR LOOP1=150 TO 189

DO
COLOR=C
C==+1 IF C>8 THEN C=1 FI
PLOT(0,LOOP1) DRAWTO(79,LOOP1)
OD

FOR LOOP1= 5 TO 70 STEP 9

DO
C=1
FOR LOOP2 =LOOP1 TO LOOP1+6
DO
COLOR=C
C==+1 IF C>8 THEN C=1 FI
PLOT(LOOP2,149) DRAWTO(LOOP2,5
3)
OD
OD

LOOP1=39 LOOP2=39 RO=0

DO
COLOR=C
PLOT(LOOP1,RO) DRAWTO(LOOP2,RO)
RO==+1 LOOP1==--1 LOOP2==+1
C==+1 IF C>8 THEN C=1 FI
UNTIL RO=30
OD

C=0

FOR LOOP1 = 35 TO 45

DO
COLOR=C
PLOT(20,LOOP1) DRAWTO(59,LOOP1)
OD

RETURN

PROC DR5()

GRAPHICS(10) COL()
LM=0 RM=79 BM=95 RO=0

DO

COLOR=C
PLOT(LM,RO) DRAWTO(RM,RO)
DRAWTO(RM,BM) DRAWTO(LM,BM)
DRAWTO(LM,RO)

PLOT(LM,95+RO) DRAWTO(RM,95+RO)
DRAWTO(RM,95+BM) DRAWTO(LM,95+B
M)
DRAWTO(LM,95+RO)

C==+1 IF C>8 THEN C=1 FI
LM==+1 RM==--1 BM==--1 RO==+1
UNTIL LM=69

OD

RETURN

PROC INTRO() ;I WILL ADD SOME
INSTRUCTIONS HERE

```

Continued on next page

```

GRAPHICS(18) ;GRAPHICS 2+16 (W
ITHOUT TEXT WINDOW)

```

```

POSITION(1,0)
PRINTDE(6,"use START to quick") ;PR
INTDE(6,"TEXT") IS THE METHOD ;TH
POSITION(1,2)
AT ACTION USES FOR THE EQUIVALENT OF B
ASIC'S
PRINTDE(6,"advance pictures or ") ;PR
INTDE(6,"TEXT"
POSITION(1,4)
PRINTDE(6,"wait and they will")
POSITION(0,6)
PRINTDE(6,"change after a short")
POSITION(2,8)
PRINTDE(6,"period of time")
POSITION(0,10)
PRINTDE(6,"press START to go on")
DO ;BE
GIN LOOP

```

```

UNTIL CONSOLE =6 OD ;AN
D WAIT TILL START IS PRESSED

```

```

RETURN

```

```

PROC MAIN() ;THIS IS THE MAIN RO
UTINE WHICH CALLS BACK TO THE DRAW ROU
TINES.

```

```

;NOTICE HOW NICE YOU
CAN FORMAT THE TEXT TO SUIT YOUR READ
ING NEEDS
INTRO()

```

```

DO
DR1() SCROLL() ;THIS CALLS THE DR1
ROUTINE WHICH CALLS COL() THEN SCROLL()
ETC.
DR2() SCROLL() ;AND 2
DR3() SCROLL() ;
DR4() SCROLL() ;
DR5() SCROLL() ;AND DOWN TO 5
OD ;CONTINUE FOREVER

```

```

RETURN

```

```

;YOU CAN ADD YOUR OWN DRAWING ROUTINES
TO THE SYSTEM BY
;JUST LABELING THEM DR# WHERE NUMBER I
S THE ROUTINE TO USE
;REMEMBER TO ADD THE ROUTINE CALL INTO
THE MAIN() ROUTINE AND CALL THE SCROL
L() ROUTINE TO GET
;THE RAINBOWING EFFECT AFTER YOU ARE F
INISHED..
;THE TIMER VALUES IN SCROLL() CAN BE I
NCREASED TO SUIT YOUR VIEWING TIMES.

```

```

PROC CSET()

```

```

;*****
;# CHARACTER SET UTILITY
;# MOVES ROM TO RAM
;# FOR CUSTOM
;# CHARACTER SET
;*****
;
;

```

```

BYTE CHBASE=756, ;POINTE
R FOR CHARACTER BASE
RAMTOP=106, ;TOP OF
FREE RAM (USUALLY 40960)
WSYNC=54282, ;WAIT S
YNC
VCOUNT=54283, ;VERTIC
AL LINE COUNTER
CO1=$D01A, ;HARDWA
RE REGISTER FOR SCREEN (GR.18)
CO2=$D016, ;HARDWA
RE REGISTER FOR 708
CLOCK=20 ;RTC AT
18,19,20

```

```

CARD ROMSET, ;VARIAB
LE FOR STANDARD SET
LOOP ;FOR LO
ADING IN CHARACTER SET
BYTE ARRAY NEWSET ;VARIAB
LE FOR NEW SET

```

```

RAMTOP==4 GRAPHICS(18) ;4 PAGE
5-1024 BYTES

```

```

CS 0 RE INITS POINTER

```

```

NEWSET=RAMTOP*256 ;SET UP
NEW START FOR CHARACTER DATA
ROMSET=CHBASE*256 ;ESTABL
ISH OLD ROM CHARACTER SET

```

```

MOVEBLOCK(NEWSET,ROMSET,1024) ;ZAPP..
COPY OLD ROM DATA TO NEW LOCATION
CHBASE=NEWSET/256 ;POINT
TO MY NEW SET

```

```

CLOSE(1) OPEN(1,"D1:FANCY",4,0);OPEN C
USTOM CHARACTER SET DATA FILE

```

```

FOR LOOP=0 TO 1024

```

```

DO

```

```

NEWSET(LOOP)=GETD(1) ;GET AN
D STORE BYTE IN NEW CHARACTER SET LOCA
TION

```

```

OD

```

```

CLOSE(1)

```

## RAM for ATARI\*

Fully Assembled • Lifetime Warranty

**48K/52K Memory Board** ..... \$89.95

For ATARI\* 400  
52K Addressable Memory  
Easy to Install

**32K Memory Board** ..... \$59.95

For ATARI\* 400 or 800

**16K Memory Board** ..... \$39.95

For ATARI\* 800

### BUILD YOUR OWN MEMORY

**48K/52K Bare Board** ..... \$30.00

**32K Bare Board** ..... \$20.00

**16K Bare Board** ..... \$10.00

**48K/52K Complete Kit** ..... \$75.00

**32K Complete Kit** ..... \$50.00

**16K Complete Kit** ..... \$30.00

Add \$4 Shipping & Handling Complete Boards Air Mail  
Add \$2 Shipping & Handling Bare Boards Air Mail  
Visa & MasterCard Accepted

\*ATARI is a trademark of Atari, Inc.  
Dealer Inquiries Welcome

**Tiny Tek, Inc.**

Route 1, Box 795  
Quinlan, TX 75474  
214-447-3025



# F-15 STRIKE EAGLE



## Put a \$20,000,000 Thrill in your Computer!!!



The F-15 STRIKE EAGLE, one of the world's most sophisticated fighters, costs the Air Force more than \$20,000,000.00 each. Now you too can strap into your ejection seat and prove how good you really are in exciting modern jet fighter combat. Fly combat missions, engage enemy aircraft, and destroy enemy ground targets from historic missions over Southeast Asia to today's defense of the strategic oil routes through the Straits of Hormuz.

F-15 STRIKE EAGLE (simulator) has all the features that makes the real EAGLE the great fighter aircraft it is — afterburners, multiple radars, air to air missiles, high explosive bombs, cannon, drop tanks, Electronic Counter Measures (ECM) flares, electronic ground tracking maps, Heads Up Display (HUD), outstanding 3-dimensional cockpit visibility, and realistic F-15 maneuverability. Your mission is to take off from your base, fly to and destroy your primary target through all the dangers of enemy territory including Surface to Air Missiles and enemy aircraft. Of course, you have to get back home again, too!!

F-15 STRIKE EAGLE is a very real simulation, accom-

plished with the guidance of real fighter pilots, and includes seven different combat missions, four skill levels, and an infinite number of exciting scenarios. F-15 will thrill and challenge you and give you the chance to prove you have the "Right Stuff" of an EAGLE fighter pilot!!

F-15 STRIKE EAGLE is available for Commodore 64, ATARI (48K), and Apple II (64K) computers. Suggested retail price is only \$34.95. Find STRIKE EAGLE at your local retailer, or call or write for MC/VISA, or COD orders. Add \$2.50 for Postage and Handling (Int'l add \$4.00USD). MD residents add 5% sales tax.

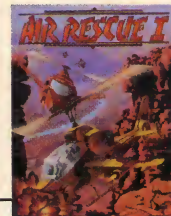
**Experience the reality of these other great simulations from MicroProse**



As close to the Real Thrill of Flying as You Can Handle!!!



Heart Pounding Accelerated Real-Time Defense of Europe Against Soviet Invasion!!!



Sensational Assault Chopper Rescue Raids for Daring Pilots!!!

## MicroProse Software

The Action is Simulated — the Excitement is REAL!!

10616 Beaver Dam Road  
Hunt Valley, MD 21030  
(301) 667-1151

Commodore 64, ATARI, APPLE, and IBM are registered trademarks of Commodore Business Machines Inc., ATARI Inc., APPLE Computer Inc., and International Business Machines Inc., respectively.



# YELLOW BRICK ROAD

by Peter Ellison

This is part two of the series called, "ROM's Glossary for the Atari." Since I will not likely have all the definitions for the Atari, I would very much appreciate hearing from whoever could add to my list.

## ROM's GLOSSARY FOR THE ATARI

**DCB:** Acronym for Device Control Block. The DCB is used by the I/O subsystem to communicate between the device handler and SIO.

**Debug:** This is where a program is tested to see if it works the way it should. If it doesn't, the programmer must go through it to find the "bug" and "de-bug it."

**Default:** A value for a parameter or variable which is supplied by a system if the user does not specify an explicit value.

**Default Mode:** On the Atari, Gr. 0 is considered the default mode, since the computer returns to it after a program is finished.

**Device Handlers:** Routines present in OS ROM which are called through ?HIO to communicate with particular devices. Currently supported are the display editor, the screen, the keyboard, the printer, and the cassette. More handlers can automatically boot in during power-up.

**Device Spec:** A special HATABS code which specifies a particular I/O device.

**Diagonal Scrolling:** This results from the combination of horizontal and vertical scrolling of the screen image. This is done by adding one to the X coordinate and one to the y coordinate or vice versa.

**Display List:** The display list specifies where the screen data may be found, what display modes to use to interpret screen data, and what special display options should be implemented.

**Display List Interrupt:** A special ANTIC display list instruction which interrupts the 6502 microprocessor during the drawing of the screen image, allowing the 6502 to change the screen parameters.

**DLI vector:** This is a 2-byte vector (low byte, high byte) to the Display List Interrupt service routine. This vector is set by the user and is located at (512,513) decimal.

**DMA:** Direct Memory Access. This happens when ANTIC halts the 6502 and takes control of the system buses to fetch an instruction or data byte from memory.

**DMACTL:** The hardware register whose bit sett-

ings control the use of DMA by the ANTIC chip. This affects, among other things, player vertical resolution, and player-missile graphics enabling.

**DOS:** Acronym for Disk Operating System which is an extension of the OS that allows the user to access disk drive mass storage as files.

**Double-Density:** The technique used to store twice as much data on a magnetic storage medium.

**Double-Line Resolution:** A unit of vertical resolution for a player in player-missile graphics. Each player byte occupies two horizontal scan lines on the screen, and each player table is 128 bytes long.

**DUP:** Acronym for Disk Utility Package. DUP is a set of utilities for disk drive usage, familiarly seen as the DOS menu. DUP executes commands by calling FMS through CIO.

**Dummy Variable:** A stand-in, which will be altered when replaced by an actual variable name or literal value.

**Dynamic Display List:** This is an ANTIC display list which the 6502 changes during vertical blank periods, allowing for even greater flexibility in the screen display.

**Edit:** To change the contents of a program or data files. Editors are classified as line-oriented editors if they work on one line at a time, and deal with text by line number.

**EOL:** In BASIC, "End-of-Line," a character with value \$9B or 155 decimal.

**Error Codes:** A statement or code printed out or displayed on the screen by a program to let you know what is happening. Error messages tell you something about what went wrong.

**Expansion Board:** A printed circuit board which accommodates components or cards that expand a computer. It is called an expansion chassis if it connects an additional cabinet to the system.

**File:** In cassette I/O, this consists of a 20-second leader of the mark tone plus any number of data bytes, and end-of-file. In diskette I/O, this consists of a number of sectors linked by pointers (125 data bytes per sector).

**File Pointer:** For diskette I/O, this is a value which indicates the current position in a file by specifying the Sector Number and the Byte Count. DOS keeps a file pointer for every file currently open.

**Fine Scrolling:** The process of horizontally or vertically scrolling a screen image in color clock or scan line increments. The horizontal scrolling or vertical

scrolling hardware registers must be used to fine scroll.

**Flag:** A status indicator for a special condition. A flag is normally stored in a flip-flop or in a register. Typically, a microprocessor provides at least the following status flags: carry, zero, sign, overflow, and half-carry or auxiliary carry.

**FMS:** File Manager System. FMS is a nonresident device handler which supports some special CIO functions.

**FONT:** A collection of characters which constitutes a character set. These characters can be either text or graphics images.

**Foreground:** Equivalent to playfield, the area of the screen which directly overlays the background of the screen. Foreground is formed by map displays and/or text.

**Format:** A resident disk handler command that clears all the tracks on a diskette.

**FORTH:** A programming language and operating system, FORTH is characterized by threaded code and postfix, or reverse Polish notation. FORTH is an expandable language; you can create new commands defined in terms of the existing commands, or in Machine language. One of the best Forths for the Atari is Valforth from Valpar International.

**Full Duplex:** A communication technique which allows data to be transmitted and received simultaneously. Most modems require you to set your computer to half duplex.

**Function:** In BASIC, a token that, when executed, returns a value to the program.

**Grid:** The invisible "checkerboard" on the TV or monitor screen that's used for text or graphics. Grids are comprised of lines of boxes arranged in columns and in rows. Grids vary as to the total number of boxes in their respective columns and rows.

**Graphics Indirection:** A special feature of the ATARI Computer which allows color register and character set generality by using indirect pointers to color and character set values.

**Half Duplex:** A mode of communication in which data may be transmitted in only one direction at a time.

**Hard Copy:** Compute output printed on paper.

**Hard Disk:** A disk composed of a magnetic coating applied to a rigid substrate, such as aluminum or ceramic. The term is generally used in contrast to "soft" (floppy) disks which are flexible. Floppy disks are slower and have less storage capacity.

**High-Resolution Graphics:** Resolution is the total number of pixels a computer uses to display a picture on screen. The higher the number of pixels, the higher the resolution; the higher the resolution, the more detail a picture can have.

**Horizontal Blank:** This is the period during which

the electron beam (as it draws the screen image) turns off and returns from the right edge of the screen to the left edge.

**Horizontal Position Register:** A special register which contains a user-definable value for the horizontal position of a player in player-missile graphics. This value is measured in units of color clocks.

**Horizontal Scan Line:** The fundamental unit of measurement of vertical distance on the screen. The scan line is formed by a single trace of the electron beam across the screen.

**Horizontal Scroll Enable Bit:** This is bit D6 of the ANTIC display instruction, which enables horizontal scrolling through HSCROL register.

**Horizontal Scrolling:** This is the process of sliding the screen window to left or right over display memory in order to display more information than could be seen with a static screen. Both coarse and fine horizontal are available.

**HSCROL:** This is the horizontal fine scrolling register located at \$D404, containing the number of color clocks by which a line is to be horizontally scrolled.

**HUE:** The upper nibble value of a color register's color. There are 16 possible hues (\$0 to \$F) which in combination with a luminance value constitute distinct colors. Examples of hues are black, red, and gold.

**IC:** Integrated circuit.

**Immediate Mode:** In BASIC, the mode where the input line is not preceded by a line number. BASIC immediately executes the line.

**Input Baud Rate:** For cassette I/O, this is assumed to be nominal 600 baud. However, this rate is adjusted by SIO to account for drive motor variations, stretched tape, etc.

**I/O:** Input/Output.

**IOCB:** Acronym for Input/Output Block. There are eight of these whose function is to communicate between the user program and CIO.

**IRQ:** Maskable (can be enabled or disabled by the 6502) interrupts such as the Break Key IRQ.

**IRQEN:** The write-only register that contains the IRQ enable/disable bits. IRQEN is shadowed at POKMSK.

**Joystick:** A popular game aid with a stick, a base and fire button. Primarily used to maneuver game shapes and graphics. Long, hard use causes fatigue in the hands and wears out the joystick.

**Kernel:** A primitive software/hardware technique which consists of a 6502 program loop which is precisely timed to the display cycle of the television set. The kernel code monitors the VCOUNT register and consults a table of screen changes catalogued as a function of VCOUNT values so that the 6502 can arbitrarily control all graphic values for the entire screen.



# Interview: Charles Bueche (Chuckles)

Interviewed by Peter Ellison

Charles Bueche (Chuckles) co-founder of Origin Systems is known for his creation of Jawbreaker II and Atari conversions of Ultima II and III. Charles and his partner, Richard Garriot (Lord British), in the past year and a half have made an impact on the computer software market.

Q. Chuck, when did you first become interested in computers?

A. It was when I was a freshman at the University of Texas.

Q. What was the first computer that you worked on?

A. It was an Apple II.

Q. And when was this?

A. In the Spring of 1980.

Q. What do you like best about the Atari Computer?

A. I like the graphics and sound.

Q. What was the first program that you converted for the Atari?

A. It was Jawbreaker II for Sierra-Online.

Q. What programs have you converted for the Atari? A. Jawbreaker II, Ultima II and III, and Caverns of Callisto.

Q. What program did you find the hardest to convert for the Atari?

A. The hardest would probably be either Ultima II or Ultima III. Ultima II was a really big program and it was early on in my conversions, but Ultima III was even bigger, and we had to delete files and stuff like that to make room.

Q. When you convert, do you do it right after the game is made, or while the game is in the programming stages?

A. I convert from finished source files.

Q. What program did you find the easiest to convert, and why? A. They all had their peculiarities, but for the easiest, it would probably be Caverns of Callisto.

Q. What, in your opinion, was your best conversion?

A. It's a toss-up between Jawbreaker II and Ultima III.

Q. When doing your conversions from the Apple to the Atari, do you ever change the game in any way?

A. I usually don't, but on Jawbreaker, I did considerable changes, and used the player/missile graphics, which helped speed up the game a lot in comparison to what I did on the Apple. But typically, I make as few changes as possible, which makes the conver-

sion a lot easier.

Q. Have you worked for any other company besides Sierra, or Origin?

A. No.

Q. In any of your programs, did you ever work with another person?

A. I consult with Richard Garriot when I do the Ultimas, and the only reason is because he wrote them, but all the rest have been my own work.

Q. Where did you get the name CHUCKLES, and have you always signed your programs with that name?

A. For the answer to part two of the question, the answer is 'yes,' I've always signed my name as Chuckles. It was a nickname that I picked up at high school, and so I figured it would be kind of appropriate for arcade game type of stuff, which I started out doing mostly.

Q. What programs have you written besides the ones you have converted to the Atari?

A. I've done Laugh Pack, Jawbreaker, Lunar Leapers and Caverns of Callisto.

Q. How long did it take you to write 'Caverns of Callisto'?

A. It took about eight months?

Q. Will there be a sequel to it?

A. I don't think so.

Q. How long did it take you to convert Ultima III?

A. It was surprisingly short, and took less time than I thought it would. From start to finish, it took about four to five months.

Q. When is Ultima IV supposed to be out?

A. We're expecting to release Ultima IV in the first quarter of '85.

Q. What do you do in your free time when you're not working on your computer?

A. I'm a car enthusiast, enjoy photography, woodworking, racquetball and snowskiing.

Q. What new programs or conversions are you working at the present time?

A. Right now I'm working on a game called 'Auto Duel,' which is based on a Steve Jackson pocket board game, 'Car Wars.' Basically, it's a computer adaptation of 'Car Wars.'

Q. And when is this supposed to be out?

A. We hope to release it in the first quarter of '85.

Q. What new games can we look forward to, coming from Origin in the next few months.

A. We've got Andy Greenburg, author of Wizardry, working on Ogre, which is another Steve Jack-

*Continued on Page 60*

# **ATARI TODAY**

— **ATARI 800XL with 1050 Disk Drive \$498.00**

**ATARI — The Winning Combination For 1985.**

See Us About

## **Computers**

ATARI 15XE

ATARI 65XEM

ATARI 130XE

ATARI 130ST

ATARI 530ST

## **Features**

Advance Music  
Capabilities

130K RAM

16 bit microprocessor

32 bit microprocessor  
Mouse  
530 K RAM  
Advanced Operating System

**Broadway Computer Center  
683 E. Broadway  
Vancouver, B.C. V5T 1X7  
872-6601**

*Call for Prices  
Full Support, Sales, and Service for ATARI.*

# AIR ATTACK

Plane Score: 103  
Ship Score : 95



By Bob Cockcroft



Air Attack is an assembly language program for two players. One player controls the movement and bombs for the plane, and the other player, the ship. The object of the game is simple. The plane bombs the ship, and the ship shoots at the plane. To avoid being hit, the plane can move left or right, and up and down. The ship can only move left and right, but can fire in three directions: Up, Left, and Right.

Each player has his or her own advantages and disadvantages. The plane is able to fly up and down, making it more difficult for the ship to shoot it. The boat, on the other hand, can fire three shots to the plane's one. This makes it easier for the boat to shoot down the plane. To destroy the plane, the boat must hit it directly in the center. The same goes for the plane. This makes it a little more difficult.

When a player is hit he blows up, and a point is added to the other person's score. There is no set limit, so play as long as you want. To pause the game, simply press the spacebar once, and the game will stop. To restart it, press any key.

This game, written on the Synassembler, requires 48K and two joysticks. It uses player/missile graphics to create the boat and plane. For the background, character graphics written in BASIC are used.

This game can be acquired by sending \$3.00 and a blank disk or cassette to:

ROM MAGAZINE  
P.O. BOX 252  
Maple Ridge, B.C.  
V2X 7G1  
CANADA

## BASIC LISTING

```

1 REM *****
2 REM *      AIR ATTACK      *
5 REM *      48K            *
7 REM *****
10 PS=0:SS=0
15 POSITION 10,10:?"loading data (1 m
in)"
20 FOR X=0 TO 2086
25 READ D
30 POKE 34000+X,D
40 NEXT X
82 ROMSET=57344:RAMT=PEEK(106)-4
84 NSET=RAMT*256
85 POKE 106,PEEK(106)-5:GRAPHICS 0
86 POKE 752,1:POSITION 13,10:?"loadin
g data"
87 FOR L=1 TO 1024:POKE NSET+L-1,PEEK(
ROMSET+L-1):NEXT L
90 FOR L1=1 TO 7
92 READ LOC
94 SET=NSET+LOC*8
95 FOR L2=0 TO 7:READ D
96 POKE SET+L2,D
97 NEXT L2
98 NEXT L1
99 POKE 756,NSET/256
110 DL=PEEK(560)+256*PEEK(561)
120 REM * LOAD INTERRUPT ROUTINE *
130 POKE DL+23,2+128:REM * IR MODE BYT
E+128 *
140 FOR X=0 TO 14
150 READ D

```

```

160 POKE 1770+X,D
170 NEXT X
180 REM * INTERRUPT ROUTINE *
210 REM * SET INTERRUPT POINTERS *
220 POKE 512,234:POKE 513,6
230 REM * ALLOW INTERRUPT *
240 POKE 54286,192
500 POSITION 30,18
505 POKE 752,1
510 ? "!!$%###$#$%"
520 POSITION 13,10
530 ? "
540 POSITION 15,18
550 ? "!!$%###$#$%"
560 POSITION 0,18
570 ? "$%$#$#$%$%$%"
600 POSITION 2,2
610 ? "  |  |  |  |  |  |  |  |  |  |
620 ? "  |  |  |  |  |  |  |  |  |  |
630 ? "  |  |  |  |  |  |  |  |  |  |
640 ? "  |  |  |  |  |  |  |  |  |  |
650 POSITION 34,2
660 ? " | / "
665 POSITION 34,3
670 ? " | \ "
675 POSITION 34,4
680 ? " | "
685 POSITION 34,5
690 ? " | \ "
700 POSITION 10,7
710 ? "Plane Score: "
720 POSITION 10,8
730 ? "Ship Score : "
900 POSITION 23,7:?" PS
910 POSITION 23,8:?" SS
1000 X=USR(34000)
1005 POKE 77,0
1010 SETCOLOR 2,11,2
1020 PS=PS+PEEK(1577)
1025 SS=SS+PEEK(1576)
1030 GOTO 900
20000 DATA 104,169,147,141,2,210,169,1
31,141,3,210,169,62,141,47,2,169,80,14
1,192,2,169,80,141
20005 DATA 193,2,169,15,141,194,2,169,
60,141,195,2,169,64,141,7,212,169,3,14
1,29,208,169,1
20010 DATA 141,30,208,162,0,169,0,157,
0,68,157,0,69,157,0,70,157,0,71,157,0,
67,232,224
20015 DATA 255,208,236,169,120,141,8,6
,24,105,8,141,9,6,173,8,6,141,0,208,17
3,9,6,141
20020 DATA 1,208,162,0,189,178,140,157
,192,68,232,224,5,208,245,162,0,189,18
3,140,157,192,69,232
20025 DATA 224,5,208,245,169,220,141,1
0,6,169,130,141,2,6,170,160,0,185,198,
140,157,0,70,200
20030 DATA 232,192,6,208,244,173,10,6,
141,2,208,169,1,141,18,6,141,37,6,169,
7,141,16,6
20035 DATA 169,50,141,19,6,169,10,141,
20,6,169,0,141,39,6,141,40,6,141,41,6,
141,26,6
20040 DATA 141,27,6,141,28,6,141,29,6,
141,32,6,141,34,6,141,35,6,141,36,6,14
1,38,6
20045 DATA 238,22,6,173,22,6,201,50,24
0,3,76,13,134,169,0,141,22,6,173,132,2
,201,0,208
20050 DATA 3,76,10,134,173,120,2,201,1
1,240,7,201,7,240,31,76,10,134,173,8,6
,201,50,240
20055 DATA 18,206,8,6,173,8,6,141,0,20
8,206,9,6,173,9,6,141,1,208,76,10,134,
173,8
20060 DATA 6,201,190,240,18,238,8,6,17
3,8,6,141,0,208,238,9,6,173,9,6,141,1,
208,76
20065 DATA 10,134,76,13,134,173,34,6,2
01,0,240,3,76,99,135,238,21,6,173,21,6
,205,20,6
20070 DATA 240,3,76,99,135,169,0,141,2
1,6,173,121,2,201,15,240,3,141,16,6,20
1,11,208,3
20075 DATA 76,100,134,201,7,208,3,76,1
29,134,201,14,208,3,76,162,134,201,10,
208,3,76,162,134
20080 DATA 201,6,208,3,76,162,134,201,

```

15,240,3,76,193,134,173,16,6,76,224,13  
 4,174,20,6,224  
 20085 DATA 255,240,19,173,39,6,201,0,2  
 40,8,224,10,240,1,202,76,123,134,232,1  
 42,20,6,76,224  
 20090 DATA 134,174,20,6,224,1,240,246,  
 173,39,6,201,0,240,8,224,250,240,1,232  
 ,76,156,134,224  
 20095 DATA 10,240,1,202,142,20,6,76,22  
 4,134,174,2,6,224,50,240,4,202,142,2,6  
 ,173,16,6  
 20100 DATA 201,10,208,3,76,100,134,201  
 ,6,208,3,76,129,134,76,224,134,174,2,6  
 ,224,175,240,4  
 20105 DATA 232,142,2,6,173,16,6,201,9,  
 208,3,76,100,134,201,5,208,3,76,129,13  
 4,76,224,134  
 20110 DATA 173,20,6,201,250,208,29,173  
 ,39,6,201,0,240,8,169,0,141,39,6,76,25  
 4,134,169,1  
 20115 DATA 141,39,6,76,58,135,141,39,6  
 ,76,58,135,173,39,6,201,0,208,3,76,17,  
 135,76,58  
 20120 DATA 135,174,2,6,160,0,185,198,1  
 40,157,0,70,232,200,192,10,208,244,173  
 ,10,6,201,225,208  
 20125 DATA 5,169,25,141,10,6,238,10,6,  
 173,10,6,141,2,208,76,99,135,174,2,6,1  
 60,0,185  
 20130 DATA 188,140,157,0,70,200,232,19  
 2,10,208,244,173,10,6,201,25,208,5,169  
 ,225,141,10,6,206  
 20135 DATA 10,6,173,10,6,141,2,208,76,  
 99,135,173,133,2,201,0,240,3,76,142,13  
 5,173,26,6  
 20140 DATA 201,0,208,26,173,10,6,141,1  
 2,6,173,2,6,141,4,6,169,1,141,26,6,173  
 ,39,6  
 20145 DATA 141,30,6,76,142,135,173,26,  
 6,201,0,208,3,76,3,136,173,30,6,201,0,  
 208,50,238  
 20150 DATA 12,6,238,4,6,174,12,6,142,4  
 ,208,172,4,6,169,0,153,0,67,185,1,67,2  
 4,105  
 20155 DATA 2,153,1,67,185,2,67,24,105,  
 3,153,2,67,185,3,67,24,105,2,153,3,67,  
 76,3  
 20160 DATA 136,206,12,6,238,4,6,174,12  
 ,6,142,4,208,172,4,6,169,0,153,0,67,18  
 5,1,67  
 20165 DATA 24,105,1,153,1,67,185,2,67,  
 24,105,3,153,2,67,185,3,67,24,105,1,15  
 3,3,67  
 20170 DATA 76,3,136,162,0,169,12,141,3  
 1,6,189,27,6,201,0,208,3,76,108,136,17  
 3,132,2,201  
 20175 DATA 0,240,3,76,47,136,173,120,2  
 ,201,11,208,3,76,53,136,201,7,208,3,76  
 ,62,136,222  
 20180 DATA 5,6,76,71,136,222,5,6,222,1  
 3,6,76,71,136,222,5,6,254,13,6,76,71,1  
 36,189  
 20185 DATA 13,6,157,5,208,188,5,6,185,  
 0,67,24,109,31,6,153,0,67,185,1,67,24,  
 109,31  
 20190 DATA 6,153,1,67,169,0,153,2,67,7  
 6,108,136,232,224,3,240,20,224,2,208,8  
 ,169,192,141  
 20195 DATA 31,6,76,10,136,169,48,141,3  
 1,6,76,10,136,173,4,6,201,196,208,23,1  
 69,0,141,26  
 20200 DATA 6,141,4,208,141,12,6,172,4,  
 6,153,1,67,153,2,67,153,3,67,162,0,189  
 ,5,6  
 20205 DATA 201,25,240,10,189,13,6,201,  
 10,240,3,76,202,136,169,0,157,27,6,157  
 ,5,208,157,13  
 20210 DATA 6,188,5,6,153,0,67,153,1,67  
 ,232,224,3,208,214,76,210,136,173,132,  
 2,201,0,240  
 20215 DATA 11,169,0,141,0,210,141,1,21  
 0,76,43,137,169,90,141,0,210,169,105,1  
 41,1,210,173,27  
 20220 DATA 6,201,0,208,5,162,0,76,21,1  
 37,173,28,6,201,0,208,5,162,1,76,21,13  
 7,173,29  
 20225 DATA 6,201,0,208,5,162,2,76,21,1  
 37,76,43,137,169,1,157,27,6,173,8,6,24  
 ,105,7  
 20230 DATA 157,13,6,169,192,157,5,6,76  
 ,43,137,173,8,208,74,144,3,76,61,137,7  
 4,144,3,76  
 20235 DATA 61,137,76,108,137,173,32,6,  
 201,0,240,3,76,108,137,169,1,141,32,6,  
 141,33,6,169

20240 DATA 0,141,26,6,141,35,6,141,4,2  
 08,141,12,6,172,4,6,153,1,67,153,2,67,  
 153,3  
 20245 DATA 67,76,108,137,162,0,189,9,2  
 08,74,74,74,176,3,76,146,137,169,1,141  
 ,34,6,169,0  
 20250 DATA 157,27,6,157,5,208,157,13,6  
 ,188,5,6,153,0,67,153,1,67,232,224,3,2  
 08,215,76  
 20255 DATA 154,137,169,1,141,30,208,76  
 ,162,137,173,32,6,201,0,208,3,76,244,1  
 38,238,35,6,173  
 20260 DATA 8,6,24,105,3,141,3,208,173,  
 33,6,201,1,208,3,76,218,137,201,2,208,  
 3,76,21  
 20265 DATA 138,201,3,208,3,76,80,138,2  
 01,4,208,3,76,139,138,76,198,138,173,3  
 5,6,201,252,208  
 20270 DATA 11,238,33,6,169,0,141,35,6,  
 76,244,138,201,1,240,3,76,244,138,169,  
 222,141,195,2  
 20275 DATA 160,192,162,0,189,208,140,1  
 53,0,71,232,200,224,9,208,244,169,24,1  
 41,0,210,169,14,141  
 20280 DATA 1,210,76,244,138,173,35,6,2  
 01,250,208,11,238,33,6,169,0,141,35,6,  
 76,244,138,201  
 20285 DATA 1,240,3,76,244,138,169,11,1  
 41,195,2,160,192,162,0,189,218,140,153  
 ,0,71,232,200,224  
 20290 DATA 9,208,244,169,32,141,0,210,  
 169,14,141,1,210,76,244,138,173,35,6,2  
 01,250,208,11,238  
 20295 DATA 33,6,169,0,141,35,6,76,244,  
 138,201,1,240,3,76,244,138,169,53,141,  
 195,2,160,192  
 20300 DATA 162,0,189,228,140,153,0,71,  
 232,200,224,9,208,244,169,100,141,0,21  
 0,169,5,141,1,210  
 20305 DATA 76,244,138,173,35,6,201,250  
 ,208,11,238,33,6,169,0,141,35,6,76,244  
 ,138,201,1,240  
 20310 DATA 3,76,244,138,169,56,141,195  
 ,2,160,192,162,0,189,238,140,153,0,71,  
 232,200,224,9,208  
 20315 DATA 244,169,60,141,0,210,169,10  
 ,141,1,210,76,244,138,169,0,141,35,6,1  
 41,195,2,141,32  
 20320 DATA 6,169,1,141,33,6,162,0,160,  
 192,189,248,140,153,0,71,232,200,224,9  
 ,208,244,169,0  
 20325 DATA 141,0,210,141,1,210,238,41,  
 6,76,244,138,173,38,6,201,0,240,3,76,8  
 7,139,173,34  
 20330 DATA 6,201,0,208,3,76,84,139,238  
 ,2,6,238,2,6,173,2,6,201,192,208,3,76,  
 81,139  
 20335 DATA 201,191,208,3,76,81,139,162  
 ,0,172,2,6,169,0,153,254,69,153,255,69  
 ,173,39,6,201  
 20340 DATA 0,208,15,189,188,140,153,0,  
 70,232,200,224,5,208,244,76,84,139,189  
 ,198,140,153,0,70  
 20345 DATA 232,200,224,5,208,244,76,84  
 ,139,76,87,139,76,168,140,173,34,6,201  
 ,0,208,3,76,168  
 20350 DATA 140,238,36,6,169,1,141,38,6  
 ,173,37,6,201,1,208,3,76,139,139,201,2  
 ,208,3,76  
 20355 DATA 198,139,201,3,208,3,76,1,14  
 0,201,4,208,3,76,60,140,76,119,140,173  
 ,36,6,201,250  
 20360 DATA 208,11,238,37,6,169,0,141,3  
 6,6,76,168,140,201,1,240,3,76,168,140,  
 169,222,141,194  
 20365 DATA 2,160,192,162,0,189,208,140  
 ,153,0,70,232,200,224,9,208,244,169,24  
 ,141,0,210,169,14  
 20370 DATA 141,1,210,76,168,140,173,36  
 ,6,201,250,208,11,238,37,6,169,0,141,3  
 6,6,76,168,140  
 20375 DATA 201,1,240,3,76,168,140,169,  
 11,141,194,2,160,192,162,0,189,218,140  
 ,153,0,70,232,200  
 20380 DATA 224,9,208,244,169,32,141,0,  
 210,169,14,141,1,210,76,168,140,173,36  
 ,6,201,250,208,11  
 20385 DATA 238,37,6,169,0,141,36,6,76,  
 168,140,201,1,240,3,76,168,140,169,53,  
 141,194,2,160  
 20390 DATA 192,162,0,189,228,140,153,0  
 ,70,232,200,224,9,208,244,169,100,141,  
 0,210,169,5,141,1  
 20395 DATA 210,76,168,140,173,36,6,201  
 ,250,208,11,238,37,6,169,0,141,36,6,76

```

,168,140,201,1
20400 DATA 240,3,76,168,140,169,56,141
,194,2,160,192,162,0,189,238,140,153,0
,70,232,200,224,9
20405 DATA 208,244,169,60,141,0,210,16
9,10,141,1,210,76,168,140,169,0,141,34
,6,141,36,6,141
20410 DATA 194,2,141,38,6,169,1,141,37
,6,162,0,160,192,189,248,140,153,0,70,
232,200,224,9
20415 DATA 208,244,169,1,141,40,6,169,
0,141,0,210,141,1,210,96,173,252,2,201
,33,240,249,76
20420 DATA 168,133,1,9,59,255,127,128,
136,184,255,254,0,1,3,63,255,28,0,0,0,
0,0,128

```

```

20425 DATA 192,252,255,56,0,0,0,0,0,
42,28,126,28,42,0,0,0,0,42,73,28,247,2
8
20430 DATA 8,34,0,0,0,24,36,67,194,36,
24,8,0,0,66,129,0,0,0,0,129,66,0
30000 DATA 1,1,5,5,21,21,85,85,85
30010 DATA 8,65,69,85,85,85,85,85
30020 DATA 3,16,80,84,85,85,85,85
30030 DATA 4,0,0,68,69,85,85,85,85
30040 DATA 5,64,64,80,84,84,84,85,85
30050 DATA 6,0,0,65,69,69,85,85,85
30060 DATA 7,0,0,8,28,62,107,8,28
31000 DATA 72,138,72,169,121,141,10,21
2
31010 DATA 141,24,208,104,170,104,64

```

## Assembler Listing

```

00010 .LI OFF
00011 .OR $8400
00012 .TA $2600
00014 .IF "D: AIRATT.0BJ"
00030 VERT0 .EQ $600
00040 VERT1 .EQ $601
00050 VERT2 .EQ $602
00060 VERT3 .EQ $603
00070 MVERT0 .EQ $604
00080 MVERT1 .EQ $605
00090 MVERT2 .EQ $606
00100 MVERT3 .EQ $607
00110 HORZ0 .EQ $608
00120 HORZ1 .EQ $609
00130 HORZ2 .EQ $60A
00140 HORZ3 .EQ $60B
00150 MHORZ0 .EQ $60C
00160 MHORZ1 .EQ $60D
00170 MHORZ2 .EQ $60E
00180 MHORZ3 .EQ $60F
00185 MIS .EQ $0300
00190 PLY0 .EQ $4400
00200 PLY1 .EQ $4500
00210 PLY2 .EQ $4600
00220 PLY3 .EQ $4700
00230 PSTICK .EQ $278
00240 PSTICK1 .EQ $279
00250 DIRSTICK .EQ $610
00260 TURCOM .EQ $611
00270 TURTY .EQ $612
00280 SPEED .EQ $613
00290 SPCOM .EQ $614
00300 PCOM .EQ $615
00310 SCOM .EQ $616
00320 EVER2 .EQ $617
00330 EVER21 .EQ $618
00340 PHF1 .EQ $626
00342 STRIG0 .EQ $284
00344 STRIG1 .EQ $285
00346 FG0 .EQ $61A
00348 FG1 .EQ $61B
00350 FG2 .EQ $61C
00352 FG3 .EQ $61D
00354 GL0 .EQ $61E
00356 STORM .EQ $61F
00358 EXF .EQ $620
00360 EFR .EQ $621
00362 PHF .EQ $622
00364 ECOM .EQ $623
00366 AECOM .EQ $624
00368 AEFR .EQ $625
00470 DECT .EQ $627
00472 PTSP .EQ $628
00474 PTSS .EQ $629
01000 ST PLA
01002 LDA H147
01003 STA $D202
01004 LDA H131
01005 STA $D203
01010 LDA H62 ;SET UP PM GRAPHICS
01020 STA $22F
01030 LDA H80
01040 STA $2C0
01050 LDA H80
01060 STA $2C1
01070 LDA H15
01080 STA $2C2
01090 LDA H60
01100 STA $2C3
01110 LDA $54000
01140 STA $D407
01150 LDA H3

01160 STA $D01D
01170 *
01190 *
01200 LDA H1
01210 STA $D01E
01220 LDA H0
01230 LDA H0
01240 L17 STA PLY0,X
01250 STA PLY1,X
01260 STA PLY2,X
01270 STA PLY3,X
01275 STA MIS,X
01280 INX
01290 J CPX $255
01300 BNE L17
01310 LDA H120
01320 STA HORZ0
01330 CLC
01340 ADC H8
01350 STA HORZ1
01360 LDA HORZ0
01370 STA $D000
01380 LDA HORZ1
01390 STA $D001
01400 *
01410 LDX H0
01420 L12 LDA SHIP,X
01430 STA PLY0+$C0,X
01440 INX
01450 CPX H5
01460 BNE L12
01470 *
01480 LDX H0
01490 L14 LDA SHIP1,X
01500 STA PLY1+$C0,X
01510 INX
01520 CPX H5
01530 BNE L14
01540 *
01560 LDA $220
01570 STA HORZ2
01580 LDA H130
01590 STA VERT2
01600 TAX
01610 LDY H0
01620 L50 LDA SHAPED,Y
01630 STA PLY2,X
01640 INY
01650 INX
01660 CPY H6
01670 BNE L50
01680 LDA HORZ2
01690 STA $D002
01700 *
01710 LDA H1
01720 STA TURTY
01725 STA AFR
01730 LDA H7
01740 STA DIRSTICK
01750 LDA H50
01760 STA SPEED
01770 LDA H10
01780 STA SPCOM
01790 LDA H0
01800 STA DECT
01805 STA PTSP
01807 STA PTSS
01810 STA FG0
01820 STA FG1
01830 STA FG2

;SET SHIP POS.
;SET TURN TYPE
;SET SPEED
;SET DIRECTION

```



01840	STA FG3	04260	* PDOWN	05630	ADC H1	06860	L400	LSR
01850	STA EXF	04270	* PDOWN	05640	STA MIS+3,Y	06870	BCC	L402
01860	STA PHF	04280	PDOWN LDX VERTZ	05650	JMP L352	06880	JMP	L401
01870	STA ECON	04290	CPX H175	05660	* MOVE SHIP MISSILES	06890	JMP	SBC
01880	STA AECON	04300	BEQ L126	05670	* MOVE SHIP MISSILES	06900	*	
01890	STA PHF1	04310	INX	05680	* MOVE SHIP MISSILES	06910	L401	LDA EXF
03000	* START SECTION	04320	STX VERT2	05690	L352 LDX H0	06920	CMP	H0
03010	* START SECTION	04330	L126 LDA DIRSTICK	05700	LDA H12	06930	BEQ	L403
03020	* START SECTION	04331	CMP H9	05710	STA STORM	06940	JMP	SBC
03030	STA INC SCOM	04332	BNE L332	05720	LL2 LDA FG1,X	06950	L403	LDA H1
03040	LDA SCOM	04333	JMP PLEF1	05730	CMP H0	06960	L403	LDA H1
03050	CMP H50	04334	JMP PLEF1	05740	BNE L353	06970	STA	EXF
03060	BEQ L190	04335	BNE L333	05750	JMP L355	06980	LDA	H0
03070	JMP PST5	04336	BNE PRIGHT	05760	L353 LDA STRIG0	06990	STA	FG0
03080	L190 LDA H0	04337	L333 JMP SHD	05770	CMP H0	07000	STA	ECON
03090	STA SCOM	04340	* SHD	05780	BEQ L356	07010	STA	SD004
03091	LDA STRIG0	04350	* TURN	05790	JMP M14	07020	STA	MH0RZ0
03092	CMP H0	04360	* TURN	05800	L356 LDA PSTICK	07030	LDY	MUERT0
03094	BNE L110	04370	* TURN	05810	CMP H11	07040	STA	MIS+1,Y
03095	JMP DRAWS	04380	SHD LDA SPCOM	05820	BNE L357	07050	STA	MIS+2,Y
03100	LL10 LDA PSTICK	04390	CMP H250	05830	JMP M11	07060	STA	MIS+3,Y
03110	CMP H11	04400	BNE L205	05840	L357 CMP H7	07070	JMP	SBC
03120	BEQ SLEFT	04410	LDA DECT	05850	BNE M14	07080	*	
03130	CMP H7	04420	CMP H0	05860	JMP M7	07090	* SHIP GUN HITS	
03140	BEQ SRIGHT	04430	BEQ L220	05870	* SHIP GUN HITS	07100	*	
03150	JMP DRAWS	04440	LDA H0	05880	M14 DEC MUERT1,X	07110	SBC	LDX H0
03160	* SHIP LEFT	04450	STA DECT	05890	JMP L360	07120	LM10	LDA SD009,X
03170	* SHIP LEFT	04460	JMP L221	05900	* SHIP GUN HITS	07130	LSR	
03180	* SHIP LEFT	04470	L220 LDA H1	05910	M11 DEC MUERT1,X	07140	LSR	
03190	SLEFT LDA HORZ0	04480	STA DECT	05920	DEC MH0RZ1,X	07150	LSR	
03200	CMP H50	04490	JMP SHC	05930	JMP L360	07160	BCS	L410
03210	BEQ L10	04500	L221 STA DECT	05940	* SHIP GUN HITS	07170	JMP	LL20
03220	DEC HORZ0	04510	JMP SHC	05950	M7 DEC MUERT1,X	07180	L410	LDA H1
03230	LDA HORZ0	04520	L205 LDA DECT	05960	INC MH0RZ1,X	07190	STA	PHF
03240	STA SD000	04530	CMP H0	05970	JMP L360	07200	LDA	H0
03250	DEC HORZ1	04540	BNE L230	05980	* SHIP GUN HITS	07210	STA	FG1,X
03260	LDA HORZ1	04550	JMP SHD1	05990	L360 LDA MH0RZ1,X	07220	STA	SD005,X
03270	STA SD001	04560	L230 JMP SHC	06000	STA SD005,X	07230	STA	MH0RZ1,X
03280	L10 JMP DRAWS	04570	* SHD	06010	LDY MUERT1,X	07240	LDY	MUERT1,X
03290	* SHIP RIGHT	04580	* SHD	06020	LDA MIS,Y	07250	STA	MIS,Y
03300	* SHIP RIGHT	04590	* SHD	06030	CLC	07260	STA	MIS+1,Y
03310	* SHIP RIGHT	04600	SHD1 LDX VERT2	06040	ADC STORM	07270	LL20	INX
03320	SRIGHT LDA HORZ0	04610	LDY H0	06050	STA MIS,Y	07280	CPX	H3
03330	CMP H190	04620	L35 LDA SHAPED,Y	06060	LDA MIS+1,Y	07290	BNE	LM10
03340	BEQ L11	04630	STA PLY2,X	06070	CLC	07300	JMP	CLCOL
03350	INC HORZ0	04640	INX	06080	ADC STORM	07310	* CLEAR COLLISIONS	
03360	LDA HORZ0	04650	INX	06090	STA MIS+1,Y	07320	* CLEAR COLLISIONS	
03370	STA SD000	04660	CPY H10	06100	LDA H0	07330	* CLEAR COLLISIONS	
03380	INC HORZ1	04670	BNE L35	06110	STA MIS+2,Y	07340	CLCOL	LDA H1
03390	LDA HORZ1	04680	CMP H225	06120	JMP L355	07350	STA	SD01E
03400	STA SD001	04690	BNE L160	06130	* SHIP GUN HITS	07360	JMP	EXPL
03410	L11 JMP DRAWS	04700	LDA H25	06140	L355 INX	07370	* EXPL0510M	
03420	* DRAWS	04710	STA HORZ2	06150	CPX H3	07380	* EXPL0510M	
03430	* DRAWS	04720	L160 INC HORZ2	06160	BEQ MAX	07390	* EXPL0510M	
03440	* DRAWS JMP PST5	04730	LDA HORZ2	06170	CPX H2	07400	EXPL	LDA EXF
03450	* PLANE	04740	STA SD002	06180	BNE H11	07410	CMP	H0
03460	* PLANE	04750	JMP NET	06190	LDA H192	07420	BNE	L411
03470	* PLANE	04760	* SHC	06200	STA STORM	07430	JMP	HPD
03480	* PLANE	04770	* SHC	06210	JMP LL2	07431	L411	INC ECON
03490	* PLANE	04780	* SHC	06220	LL1 LDA H48	07432	LDA	MH0RZ0
03500	PST5 LDA PHF	04790	* SHC	06230	STA STORM	07433	CLC	
03510	CMP H0	04800	* SHC	06240	JMP LL2	07434	ADC	H3
03520	BEQ PST5	04810	SHC LDX VERT2	06250	* MAX	07435	STA	SD003
03530	JMP NET	04820	L206 LDA SHAPEC,Y	06260	* MAX	07440	LDA	EFER
03540	PST5 INC PCOM	04830	INX	06270	MAX LDA MUERT0	07450	CMP	H1
03550	LDA PCOM	04840	INX	06280	CMP H196	07460	BNE	L412
03560	CMP SPCOM	04850	INX	06290	BNE L361	07470	JMP	FRI
03570	BEQ L170	04860	INX	06300	LDA H30	07480	L412	CMP H2
03580	JMP NET	04870	CPY H10	06310	LDA H30	07490	BNE	L413
03590	L170 LDA H0	04880	BNE L206	06320	STA FG0	07500	JMP	FR2
03600	STA PCOM	04890	LDA HORZ2	06330	STA SD004	07510	L413	CMP H3
03610	LDA PCOM	04900	CMP H25					

03660	BEQ L150	04910	BNE L207	06340	STA MHORZ0	07520	BNE L414
03670	STA DIRSTICK	04920	LDA H225	06341	LDY MUERT0	07530	JMP FR3
03680	L150 CMP H11	04930	STA HORZ2	06342	STA MIS+1,Y	07540	L414 CMP H4
03690	BNE L100	04940	L207 DEC HORZ2	06343	STA MIS+2,Y	07550	BNE L415
03700	JMP PLEF1	04950	LDA HORZ2	06344	STA MIS+3,Y	07560	JMP FR4
03710	L100 CMP H7	04960	STA SD002	06350	*	07560	L415 JMP FR5
03720	BNE L101	04970	JMP NET	06360	L361 LDX H0	07580	*
03730	JMP1 PRIGHT	04980	*	06370	L363 LDA MUERT1,X	07590	FR1 LDA ECOM
03740	L101 CMP H14	04990	*	06380	CMP H25	07650	CMP H252
03750	BNE L19	05000	*	06390	BEQ LL400	07660	BNE L417
03760	JMP PUP	05010	NET LDA STRIG1	06392	LDA MHORZ1,X	07670	INC EFR
03770	L19 CMP H10	05020	CMP H0	06394	CMP H10	07680	LDA H0
03780	BNE L20	05030	BEQ L350	06395	BEQ LL400	07690	STA ECOM
03790	JMP PUP	05040	JMP MOV	06397	JMP L362	07700	JMP HPD
03800	L20 CMP H6	05050	L350 LDA FG0	06400	LL400 LDA H0	07710	L417 CMP H1
03810	BNE L21	05060	CMP H0	06410	STA FG1,X	07720	BEQ L418
03820	JMP PUP	05070	BNE MOV	06420	STA SD005,X	07730	JMP HPD
03830	L21 CMP H15	05080	LDA HORZ2	06430	STA MHORZ1,X	07740	L418 LDA H222
03840	BEQ L22	05090	STA MHORZ0	06432	LDY MUERT1,X	07750	STA \$2C3
03850	JMP PDOWN	05100	LDA VERT2	06433	STA MIS,Y	07760	LDY H192
03860	L22 LDA DIRSTICK	05110	STA MUERT0	06434	STA MIS+1,Y	07770	LDX H0
03870	JMP SHD	05120	LDA H1	06440	L362 INX	07780	L419 LDA FRAME1,X
03880	*	05130	STA FG0	06450	CPX H3	07790	STA PLY3,Y
03890	* PLEF1	05140	LDA DECT	06460	BNE L363	07800	INX
03900	*	05150	STA GL0	06470	JMP SPF	07810	INX
03910	PLEF1 LDX SPCOM	05160	JMP MOV	06480	*	07820	CPX H9
03920	CPX H255	05170	*	06490	* SHIP TRIGGER	07830	BNE L419
03930	BEQ L120	05180	MOV LDA FG0	06500	*	07832	LDA H24
03940	LDA DECT	05190	CMP H0	06510	SPF LDA STRIG0	07833	STA SD200
03950	CMP H0	05200	BNE L354	06520	CMP H0	07834	LDA H14
03960	BEQ L300	05210	JMP L352	06530	BEQ L370	07835	STA SD201
03962	CPX H10	05220	L354 LDA GL0	06531	LDA H0	07840	JMP HPD
03964	BEQ L305	05230	CMP H0	06532	STA SD200	07850	*
03970	DEX	05240	BNE LL353	06533	STA SD201	07860	FR2 LDA ECOM
03980	L305 JMP L301	05250	INC MHORZ0	06540	JMP COL	07870	CMP H250
03990	L300 INX	05260	INC MUERT0	06550	L370 LDA H90	07880	BNE L717
04000	L301 STX SPCOM	05270	LDA MHORZ0	06551	STA SD200	07890	INC EFR
04010	L120 JMP SHD	05280	STX SD004	06552	LDA H105	07900	LDA H0
04020	*	05290	LDY MUERT0	06553	STA SD201	08000	STA ECOM
04030	*	05300	LDA H0	06559	LDA FG1	08010	JMP HPD
04040	*	05310	STA MIS,Y	06560	CMP H0	08020	L717 CMP H1
04050	PRIGHT LDX SPCOM	05320	LDA MIS+1,Y	06570	BNE L372	08030	BEQ L718
04060	CPX H1	05330	CLC	06580	LDX H0	08040	JMP HPD
04070	BEQ L120	05340	ADC H2	06590	JMP FM	08050	L718 LDA H11
04080	LDA DECT	05350	STA MIS+1,Y	06600	L372 LDA FG2	08060	STA \$2C3
04090	CMP H0	05360	LDA MIS+2,Y	06610	CMP H0	08070	LDY H192
04100	BEQ L303	05370	CLC	06620	BNE L373	08080	LDX H0
04102	CPX H250	05380	ADC H3	06630	LDX H1	08090	L719 LDA FRAME2,X
04104	BEQ L320	05390	STA MIS+2,Y	06640	JMP FM	08100	STA PLY3,Y
04110	INX	05400	LDA MIS+2,Y	06650	L373 LDA FG3	08110	INX
04120	L320 JMP L302	05410	CLC	06660	CMP H0	08120	INX
04130	L303 CPX H10	05420	ADC H2	06670	BNE L374	08130	CPX H9
04132	BEQ L302	05430	STA MIS+3,Y	06680	LDX H2	08140	BNE L719
04134	DEX	05440	JMP L352	06690	JMP FM	08142	LDA H32
04140	L302 STX SPCOM	05450	*	06700	L374 JMP COL	08143	STA SD200
04150	L121 JMP SHD	05460	LL353 DEC MHORZ0	06710	*	08144	LDA H14
04170	* PUP	05470	INC MUERT0	06720	FM LDA H1	08145	STA SD201
04180	*	05480	LDA MHORZ0	06730	STA FG1,X	08150	JMP HPD
04190	PUP LDX VERT2	05490	STX SD004	06740	LDA HORZ0	08170	FR3 LDA ECOM
04200	CPX H50	05500	LDY MUERT0	06742	CLC	08180	CMP H250
04210	BEQ L125	05510	LDA H0	06744	ADC H7	08190	BNE L720
04220	DEX	05520	STA MIS,Y	06750	STA MHORZ1,X	08200	INC EFR
04230	STX VERT2	05530	LDA MIS+1,Y	06760	LDA H192	08210	LDA H0
04240	L125 LDA DIRSTICK	05540	CLC	06770	STA MUERT1,X	08220	STA ECOM
04241	CMP H10	05550	ADC H1	06780	JMP COL	08230	JMP HPD
04242	BNE L330	05560	STA MIS+1,Y	06790	*	08240	L720 CMP H1
04243	JMP PLEF1	05570	LDA MIS+2,Y	06800	* COLLISIONS	08250	BEQ L721
04244	L330 CMP H6	05580	CLC	06810	*	08260	JMP HPD
04245	BNE L331	05590	ADC H3	06820	COL LDA SD008	08270	L721 LDA H53
04246	JMP PRIGHT	05600	STA MIS+2,Y	06830	L5R	08280	STA \$2C3
04247	L331 JMP SHD	05610	LDA MIS+3,Y	06840	BCC L400	08290	LDY H192
04250	*	05620	CLC	06850	JMP L' L401	08300	LDX H0

Continued on Page 62



# Synapse: A Business Approach

by Peter Ellison

This article, originally called "Atari: Taking Care of Business," was changed to "Synapse: A Business Approach," because of the sale of Atari to Jack Tramiel. Atari had originally agreed to buy Synfile+, Syn-trend, and SynCalc from Synapse, but that since has fallen through, forcing Synapse to market their own products. These three programs, designed by Synapse, are completely compatible with one another, thus making it easy to merge files for different applications. These programs are all in attractive binders and have clear concise instructions for each command. Inside the front cover of each is a fold-out reference card. It is made of sturdy cardboard, stands by itself, and is, therefore, easy to use when doing such things as typing.

The first program that I have used extensively is "Synfile+." I had previously been using a three-year old program called "Filemanager+," which is also from Synapse, but is a lot slower because some of it was written in Basic. I decided to convert my entire filing system to Synfile+. I thought it would be slow and difficult, but to my surprise, it was fast and easy; therefore, I was finished in no time at all.



Once you have the program up and running, the main menu will appear on your screen, giving three different options. They are: Files, Records, or Reports. Choosing the first option will bring you to another menu with eleven different sub commands which have to do with anything pertaining to working with your files, such as OPENing, CLOSEing, or MERGEing a file.

The next option in the main menu is Records. Choosing this will bring you to another menu with five choices. To be able to use this option from the

main menu a file must first be loaded. The selection includes the ability to enter, update, delete, re-index, or retrieve a record. One of the features that really impresses me is the ability to enter a record that is 80 characters long, all on the same line. This is done by scrolling the screen so that you know how it will actually look. Another feature is how the program keeps track of the number of records that are entered in the top right-hand corner of the screen.

The ability to retrieve a record quickly simplifies the making of corrections. Calculations can also be made very easily because of the menu driven prompts.

The last option in the main menu is Reports. When picking this, a menu with only two choices will appear on the screen. They are LISTS, and LABELS. The first choice, LISTS, is a free-form where you get to design the actual layout. The second, LABELS, is like a columnar table. When choosing Lists, a numbered bar will appear on the bottom of the screen. This is what you use to decide how long or short you want each field to be.

Special features in this package make this product quite valuable. One of these is the tutorial disk that comes along with the binder, taking you step by step through the program. This makes it perfect for first time data base users. There are many features that I still haven't used, but I know that they will be there whenever I need them. I still have not used it to make any calculations, but have tested it out, and it works fine. Synfile+ is a must for anyone wanting to keep records for any small business or home, having the ease of use with much power.

The second program, "SynCalc," is an easy to use electronic spreadsheet program that has many uses in both the home and office. SynCalc's personal applications might include calculating income taxes, setting up a personal budget, or balancing a checkbook. SynCalc also has many business applications including sales projections, financial ratios, engineering analysis, or cost estimates. This program functions like a business or scientific worksheet, combining the convenience of a pocket calculator with the powerful memory and electronic screen capabilities of the personal computer.

After using this program I can see that it was designed for the first-time user as well as for the experienced spreadsheet user. SynCalc's menu driven interface guides the first-time user through the opera-

tions in a step-by-step manner, while the corresponding expert user commands are echoed at the top of the screen. In this way, the beginner can quickly learn the expert user commands, and will, in no time, be using this program like a pro. On the other hand, the experienced user can enter the expert user command mode to efficiently execute the desired functions.

The sheet itself has been organized as a grid of columns and rows. These intersect and define thousands of entry positions for data. Each position can take an entry of an alphabetic label, a number, or a formula to be calculated. You can individualize the appearance of each entry by formatting the sheet according to your own specifications. You can, for



example, set up your income tax calculations to look like the forms you are using. I found this quite handy because I've always had a little trouble filling out my forms.

SynCalc will remember the formulas and calculations you are using to work through a problem. You can then go back at any time and change a previously entered number, and it will automatically recalculate all the relevant formulas, and display the new results. This feature makes it a powerful planning and forecasting tool since it allows you to experiment with, and examine a number of alternatives. For example, you might want to know how fluctuating sales figures can affect your company. This testing of alternatives is called "what if" analysis, and is one of the major applications of spreadsheet programs.

SynCalc allows you to change labels, numbers, or formulas in any of the rows and columns on the worksheet, and will instantly restructure the worksheet to reflect these changes. Formulas, numbers, and labels can be copied from one position to any number of successive positions on the worksheet. You can also sum, average, or otherwise manipulate rows, columns, or other ranges of cells. The space in which titles and numbers are displayed can be shortened or lengthened without affecting how they are stored in memory. Finally, your electronic worksheet can be saved to disk and printed in whole or part.

Because of the ease and flexibility of the easy cardboard reference sheet, I was using SynCalc in an hour, and I was able to calculate all my friends' income tax forms, (for a fee of course), so this program can also be a money maker for you.

The third program, "SynTrend," is actually two programs in one. They are called "SynStat" and "SynGraph" which are contained on two separate disks.

Synstat is a statistical program for quickly and easily analyzing financial trends for your home, business, or schoolwork. Synstat is a powerful forecasting and planning tool. With it you can make analyses on variables, perform simple and multiple regressions, and write out files for graphing. Synstat can be converted to the Data Interchange Format (DIF) which allows them to be displayed with SynGraph.

SynGraph, on the other hand, is used to create clearly labeled, high resolution, color-coded graphs from data entered from SynGraph, SynStat, SynFile+, SynCalc, or Visicalc. You can choose from four commonly used graphing methods: line graph, bar graph, scatter plot, or pie chart, to represent your data. You can rescale and relabel a graph, display up to three factors at a time on the same graph, save the graphs for viewing later, or output to a compatible printer. One feature I enjoyed was the ability to make your graphs into a slide show that you could present to



either business associates or friends.

Every one of these programs comes complete with a tutorial inside its booklet, outlining each function. This makes the program easy to understand and use. Also, inside each plastic binder is a place to put your disk to keep it safe. I recommend each of these programs if you have a small business, or a lot of personal records that you want to keep track of. Also these programs will prevent anyone from saying that all you do is play games on your Atari, you can now perform some practical home or business functions.

# Atari's Timing System

by Bob Cockroft

This article is the first in a series on the Atari's timing system. It will be the goal of this series to explain in some detail how the timing system operates, and to present a number of practical applications. This first article will give an overview of the system and will apply its basic principles to the screen COLOR Register.

Many people program without using the timing system. This is unfortunate in that the Atari has an effective system that can be used in a number of ways. In fact, the time base which the circuitry operates is synchronized with the television signal. This produces both a distinct graphics screen and some interesting programming possibilities. There are 2 main types of standard television signals(NTSC/PAL). Canada and the United States use the NTSC system which creates 60 frames per second and 262 lines per frame. However, European countries use the PAL system which creates, instead, 50 frames per second and 312 lines per frame. As a result of the difference, the Atari must modify its timing system to make it compatible with either the NTSC or PAL standards.

## The PAL byte

The standard(NTSC/PAL) your television uses can be determined by examining the 'PAL' byte(\$D014/dec 53268). If the PAL byte equals zero, the PAL standard is used. Conversely, if the PAL byte does not equal zero, your television uses the NTSC standard.(see below) For simplicity, this article assumes that you have a NTSC standard television.

PAL \$D014

dec. 53268

bit number:

b0

...	...	...	...	...	...not...
..not	used...	b3...	b2...	b1...	used...
NTSC	=	1	1	1	= 13
PAL	=	0	0	0	= 0

Every 1/60th of a second the electron beam of a NTSC standard television creates a new screen by drawing horizontal lines of graphics from top to bottom. The 6502 processor synchronizes with the televi-

sion by causing a system interrupt after every screen is drawn(per. of 1/60 of a sec.). The period in which the electron beam of the T.V. returns to the top of the screen from the bottom is called the VERTICAL BLANK. The computer interprets this period as a non-maskable interrupt. As a result, an interrupt occurs regularly every 1/60th of a second. This interrupt, in conjunction with the Timers, can be used to cause the computer to automatically change colours, sound, or any other function.

By setting its system clock rate to 1.79 MHz, the distance in which the electron beam moves across the screen can be measured in terms of machine cycles. For example, the time in which the 6502 executes a 4 cycle instruction like 'EOR'(absolute), the electron beam will move 2 character widths across the screen. As a result, colour changes or many other types of graphics effects can be done in the middle of a horizontal line of graphics. Unfortunately, because the timing system is not exact(in terms of small fractions of a second) it is often difficult to pinpoint the precise spot on the screen where you want the graphics effect to occur.

The Atari computer has 6 timers. Their address and interrupt vectors are listed below.

Symbol (Timer)	Location (dec.),	Symbol Vec/Flag	Interrupt Vector/flag
RTCLOCK	18,19,20		none
CDTMV1	536,537	CDTMA1	550,551
CDTMV2	538,539	CDTMA2	552,553
CDTMV3	540,541	CDTMA3	554
CDTMV4	542,543	CDTMA4	556
CDTMV5	544,545	CDTMA5	558

The first item on the above list, Realtime Clock(RTCLOCK), is the most convenient timer; therefore, it should be used where simple timing routines are needed. This device uses 3 bytes from 18 to 20 dec. The first byte, address 20 increments every VBLANK. In other words, once every 1/60th of a second the value in this location increases by one. When address 20 reaches a value of '255', it is reset to '0' and address 19 is incremented by one. In turn, when address 19 reaches a value of '255' it is reset to '0' and address 18 is incremented by one.(see below)



## Realtime Clock

			+1 < 255
LOC.	18	19	20
			+1 < 255

Below is a program which uses these bytes to make a basic clock.

```

10 REM *****
20 REM *
25 REM * REALTIME CLOCK *
30 REM *
35 REM *****
50 GRAPHICS 0
52 REM * TURN OFF CURSOR *
55 POKE 752,1
58 REM * SET REALTIME CLOCK TO ZERO *
60 POKE 18,0:POKE 19,0:POKE 20,0
70 POSITION 17,8:?"CLOCK"
80 POSITION 16,12:?"Hr Min Sec"
90 TH=0:TM=0:TS=0
97 REM *
98 REM * REFER TO REALTIME CLOCK *
100 T=INT(PEEK(19)*4+PEEK(20)/60)
101 REM * DETERMINE SEC,MIN AND HOURS *
102 TS=T
104 IF T>59 THEN TS=0:TM=TM+1:POKE 19,0:POKE 20,0
107 IF TM>59 THEN TH=TH+1
110 POSITION 16,10
120 ? " ";TH;" ";TM;" ";TS;" "
140 GOTO 100

```

The Realtime clock bytes can be changed through the POKE command (or the equivalent) to any starting value. In fact, the above program does this at line 60, where it initially sets the clock to zero. Unfortunately the Realtime clock does not have interrupt capabilities. In other words, it is unable to stop the regular functions of the Operating System when the clock reaches a predetermined point. Without an interrupt capability, a timer program is unable to take over control of the computer when another program is currently executing. Fortunately, some of the other timers have interrupt capabilities.

The Atari computer contains 5 countdown timers. Using a 2 byte configuration, these timers decrement from a user defined starting value that can range anywhere from 1 to 65536. If a value of '250' were POKED into one of these timers, it would be reduced by '1' every VBLANK(1/60 of a sec). When this value reaches zero, control is forced through the address stored in its corresponding interrupt vector. By pointing the interrupt vectors to the location of a user created machine code, most any operation can be performed even though the computer is busy doing something else. For example, suppose you wanted the computer to automatically change the screen colour while you were programming in BASIC. The first step in accomplishing this would be to make a machine language subroutine that would change the colour of

the screen. This subroutine would need to store different values to playfield color register(\$2C6)(710 dec). In addition, a RTS(return from subroutine) instruction would need to be placed at the end of the routine so that control would be given back to the Operating System. Although this code could be stored anywhere in free RAM, for the purposes of this demonstration assume that a group of bytes beginning at 1536(\$600 hex) are used. The following is a routine containing all the characteristics mentioned above.

## Machine Subroutine

dec	command	;explanation
173	LDA \$64B	;load colour value
75	lo byte	
6	hi byte	
24	CLC	;clear carry bit
105	ADC #10	;add 10 to the color value
10	amount	
141	STA \$64B	;store colour value
75	lo byte	
6	hi byte	
141	STA \$2C6	;display new colour
198	lo byte	
2	hi byte	
96	RTS	;return from subroutine

## PARTS/SERVICE FOR ATARI COMPUTERS

ORIGINAL FACTORY PARTS FOR 800/400, 810, 820, 850,  
UPGRADE TO GTIA. 48K AND  
REV. "B" OPERATING SYSTEM  
CUSTOM 810 DISK DRIVES ... \$215.00

### INTEGRATED CIRCUITS FOR 800/400

GTIA Chip ...  
upgrade with instructions ... \$11.50  
10K rev. "B" O.S. Upgrade ...  
3 chip rom set with instructions ... \$12.50  
Pokey Chip ... C012294 ... \$8.50  
Antic Chip ... C012296 ... \$10.00  
PIA Chip ... C014795 ... \$11.00

### BOARD SETS. BUILD YOUR OWN COMPUTER, DISK DRIVE

800 ... 0K ... \$72.50  
800 ... 48K ... \$135.00  
400 ... 0K ... \$52.50  
810 Board Set ... \$110.00  
All Boards Complete With IC's Etc.  
Keyboards not included.

### MODULES/CIRCUIT

#### BOARDS...complete with IC's

16K Ram Memory Module ...  
CX853 ... \$24.50  
800 10K Rev. "B" O.S. Module ... \$18.50  
800/400 CPU Board with GTIA ... \$24.50  
800 Main Board ... \$28.50  
400 Main Board ... \$26.50  
400 Main Board W/O IC'S ... \$8.50  
800 Power Supply Board ... \$10.50  
810 Data Separator Board ...  
upgrade with instructions ... \$25.00  
810 Side Board W/O Sep. & 1771 ... \$43.50  
810 Rear Power Board ... \$25.00  
810 Analog Board ... \$16.00

Overseas customers ... ask  
for PAL price list.

### MISC.

810 Rear Board/Analog Board Upgrade ...  
with 10 pin jumper and instructions ... \$39.50  
Editor Assembler ... \$29.95  
Basic Cartridge W/O Case, Manual ... \$23.50  
Cartridge Circuit Boards ... \$4.00  
Non-Atari Cartridge Boards ... \$2.00  
800/400 Repair Manual ... \$39.95  
Non-Atari power transformer ... \$16.50

### AMERICAN TV PHONE 415-352-3787

Mail Order Address ... 15338 Inverness St., San Leandro, CA 94579.  
Retail Store ... 1988 Washington Ave., San Leandro, CA 94577.

Terms: We accept money orders, personal checks or COD. VISA Mastercard OK on orders over \$25.00. No personal checks on COD.

Shipping: \$6.00 Shipping and handling on orders under \$150.00. Add \$2.00 for COD orders. CA res. include 6-1/2% sales tax. Overseas shipping extra.

Prices subject to change without notice. We reserve the right to limit quantities. Sales limited to stock on hand. AK, HI, FPO-APO, add \$5.00 on all orders.

Much More! Send SASE for free price list.

Repair and upgrade services available Call. \*Atari is a registered trademark of Atari, Inc.

The second step is to decide the length of time for which the timer is to be set. Because the timer can have a value as high as 65536 and decrements every 1/60th of a second, the maximum period is 18.2 minutes. However, for simplicity, assume that timer 2 is given a value of '100'dec(538,539).(see below)

100 dec

location 538 = 100      The 2 bytes of  
location 539 = 0      Timer 2

The computer would then begin to decrement the value (100) every VBLANK (60th of a sec). When the value reaches zero, control is passed to the address specified by Timer 2's interrupt vector bytes (552,553). To cause the colour of the screen to change, store the starting address of the machine subroutine to the interrupt vectors. As a result, when timer 2 counts down to zero, control is passed to the subroutine where the screen colour is changed. The first program at the end of the article changes the screen colour in a manner similar to the one described.

#### The SETVBV Routine

Because the count-down timers are decremented during the VBLANK process, special care is needed to set them correctly. The easiest method of setting them is to use the SETVBV routine (\$E45C). By storing the timer number in the Accumulator ('A'), the low byte of the timer value in the 'Y' Register, and the corresponding high byte in the 'X' register, 'jumping' to the SETVBV routine will set the timer automatically.

LDA #2            ;Set count-down timer 2  
LDY #250        ;low byte 250 cycles  
LDX #0          ;hi byte  
JSR SETVBV      ;jump to the routine

SETVBV = \$E45C

The following program changes screen colour in much the same way as described in this article.

```
10 REM *****
20 REM *
30 REM *      AUTOMATIC      *
40 REM *  COLOUR MODIFICATION  *
50 REM *      PROGRAM 1      *
```

```
60 REM *
70 REM *****
80 REM *
95 REM *  STORE SUBROUTINE 1(at 15
36)*
100 FOR X=1536 TO 1536+10
105 READ D
110 POKE X,D
115 NEXT X
117 REM *  STORE SUBROUTINE 2(at 1
580)*
120 FOR X=1580 TO 1580+14
125 READ D
130 POKE X,D
135 NEXT X
138 REM *
140 REM *  SET TIMER POINTERS TO *
142 REM *  SUBROUTINE 2 (1580)  *
145 POKE 552,44:POKE 553,6
147 REM *
150 X=USR(1536)
9990 REM *
9992 REM *
9995 REM *  SUBROUTINE 1 (set time
r 2) *
10000 DATA 104,169,2,160,10,162,0
,32,92,228,96
10002 REM *
10004 REM *
10005 REM *  SUBROUTINE 2 (change
colour) *
10100 DATA 173,75,6,24,105,10,141
,75,6,141,198,2
10115 REM *  RETURN TO THE FIRST S
UBROUTINE *
10117 REM *  IN ORDER TO RESET TIM
ER 2      *
10120 DATA 76,1,6
```

```
10 REM *****
20 REM *
30 REM *  SOUND DEMO      *
40 REM *  PROGRAM 2      *
50 REM *
60 REM *****
100 P60=100
110 P61=100
120 REM *  DISPLAY AND PRINT SCREE
N *
130 GRAPHICS 0
```

```

140 POSITION 13,2:? "SOUND TEST"

150 POSITION 2,20:? "Use the JOYS
TICK to change the values"
160 POSITION 2,21:? "stored in ad
dresses 53760 and 53761"
170 POSITION 2,4
180 ? "SYMBOL:      AUDF(1)      AU
DC(1)"
190 POSITION 2,15:? "AUDF(1)=Audi
o channel one frequency"
200 POSITION 2,16:? "AUDC(1)=Audi
o channel one control"
210 POSITION 2,5
220 ? "ADDRESS:      53760      53
761"
230 POSITION 2,7:? " Value:"
240 POKE 752,1
250 REM *
260 REM * USE JOYSTICK TO CHANGE
SOUND *
270 ST=STICK(0)
280 IF ST=14 AND P60<255 THEN P60
=P60+1
290 IF ST=13 AND P60>0 THEN P60=P
60-1
300 IF ST=11 AND P61<255 THEN P61

```

```

=P61+1
310 IF ST=7 AND P61>0 THEN P61=P6
1-1
320 REM *
330 REM *
340 REM * CHANGE SOUND FREQUENCY
*
350 POKE 53760,P60
360 REM *
370 REM * CHANGE VOLUME AND DISTO
RTION *
380 POKE 53761,P61
390 REM *
400 REM *
410 POSITION 14,7
420 ? P60;"          ";P61;"
"
430 GOTO 270

```

\*\*\*\*\*  
**\*800\* ATARI BASIC PROGRAMMERS \*800XL\***  
 WOULD YOU LIKE MORE UTILITY? ARE YOU TIRED OF BASIC EDITING LOCKUPS?  
 NOW FOR ONLY \$14.95 YOU CAN HAVE A GREATLY ENHANCED BASIC FOR YOUR ATARI

ENHANCEMENTS TO BASIC is a disk based machine language program which combines with ATARI BASIC. It creates an enhanced BASIC which, once created, loads automatically and does not need the BASIC cartridge or built-in 800XL BASIC.

ENHANCEMENTS TO BASIC requires that you have a disk drive, ATARI DOS 2.0B, 48K minimum memory, and either the ATARI BASIC cartridge or an 800XL with built in BASIC.

\*\*\*\*\*  
 IF YOU PROGRAM ON AN ATARI 800 USING THE BASIC CARTRIDGE, ENHANCEMENTS TO BASIC WILL ELIMINATE THE EDITING LOCKUP PROBLEM WHICH HAS PLAGUED PROGRAMMERS FOR SO MANY YEARS!  
 \*\*\*\*\*

ENHANCEMENTS TO BASIC makes over twenty direct mode (and even some program mode) commands available to you to ease your programming efforts.

Most of the DOS commands commonly used by BASIC programmers (DIRECTORY, LOCK, UNLOCK, DELETE, RENAME, and FORMAT) are available from BASIC with no need for the DOS menu.

Even though ENHANCEMENTS TO BASIC resides in RAM instead of ROM, it is safe from SYSTEM RESET and also from pokes into the area of memory where it resides.

TRON and TROFF will allow you to trace the flow of your program for debugging. By using the DUMP command you can even have your program trace printed to your printer while your screen continues to show the program graphics. You can have the trace print line numbers only, or have the entire line printed as it is executed to enable you to follow the program flow more easily. TRON and TROFF are even available in program mode so that you can have the program itself start and stop the trace at selected points by simply inserting the appropriate program line.

ENHANCEMENTS TO BASIC is priced at \$14.95 U.S. (\$21.95 Canadian) including shipping and handling. C.O.D. orders add \$2.00 additional. VISA and MASTERCARD accepted with no surcharge. Remittance in U.S. funds would be appreciated, but Canadian funds will be accepted if more convenient to you, our valued customer. ENHANCEMENTS TO BASIC may be ordered by writing to FIRST BYTE, P.O. BOX 32, RICES LANDING, PA U.S.A. or by calling 412-627-3596 Monday - Friday from 9:00 AM to 4:00 PM EST.

RENUMBER will renumber your programs starting at the line number you choose and using whatever increment you choose. It will renumber all line references except variables, and will print a list of line numbers containing variable GOTO and GOSUB etc. statements to either the screen or printer.

PROTECT saves your program in an executable but unreadable format to protect your programming secrets.

\*\*\*\*\*  
 RESTORE WILL ENABLE YOU TO RESTORE TO A NORMAL CONDITION ANY ACCIDENTALLY DELETED FILE WHICH HAS NOT BEEN WRITTEN OVER BY NEW DATA!  
 \*\*\*\*\*

Error messages are printed as standard English phrases as well as error numbers.

DELETE will delete a range of line numbers. (QUICKLY!!)

VAR will print a list of the variables used in your program to either the screen or the printer to help you find unused variable names which may have crept into your program.

Several other commands are also available to enhance your programming capabilities. Also, several bugs have been fixed in the cartridge version of ATARI BASIC.

WE FEEL CERTAIN THAT YOU WILL BE PLEASED WITH ENHANCEMENTS TO BASIC AND WE MAKE THIS GUARANTEE TO YOU:

IF YOU DO NOT FEEL THAT ENHANCEMENTS TO BASIC IS WORTH THE PRICE YOU PAID, CONTACT US WITHIN 30 DAYS AND WE WILL ARRANGE FOR A FULL REFUND OF YOUR PURCHASE PRICE.

ATARI is a trademark of ATARI, Inc., Sunnyvale, California.

\*\*\*\*\*



# Books On The Shelf

by Peter Ellison

I have owned my computer since the beginning of Atari, and have seen a great deal of change. One of the greatest changes is to that of the reading material on learning to program. The book, which you received with your Atari was good for about a day, but it offered nothing beyond instruction on BASIC and wasn't too helpful to someone who wanted to further his or her computer education. I have finally found a book that deals with machine language game programming, specifically for the Atari. The name of the book is "Atari Graphics and Arcade Game Design" by Jeffrey Stanton and Dan Pinal.

Jeffrey Stanton received a BME (1967) and a MSME (1969) from Rensselaer Polytechnic Institute. He worked as a control systems engineer and mechanical engineer for the aerospace industry in the early 1970's. His interest in computer game design side-tracked his career as a photographer and book illustrator in the late 1970's. In addition to writing several Apple arcade games, and doing some occasional consulting, he is the author of "Apple Graphics and Arcade Design," and is one of the editor/reviewers for the books of Apple and Atari Computer Software.

Dan Pinal, on the other hand, is typical of many of the early computer hobbyists: he is self educated. He was one of the first to own an Atari computer, and entered the micro-computer industry a year later. Dan consulted, taught, and did game programming for two software houses at the peak of the game market in 1983. He has one Atari game currently on the market.

These two authors are ones that know what they're talking about, and after reading the book, you'll see what I mean. If this book had come out three years earlier, there would be a lot more games on the market, but I think, with it now out, there will soon be another onslaught of Atari games. I usually don't become too excited about a new book for the Atari because most of them are made up of material that has already been written about in magazines, but this book is something truly original.

The first chapter, which is labelled "Graphic Modes and Color Registers," is one that goes through every graphic mode and color, and gives a few self explanatory programs. It also includes a large two page table that has a listing for every graphic mode, and what its function is. The columns are graphic mode, Antic mode, Available colors, Screen size, Scan Lines mode, Bytes/line, Memory Used, Color

register numbers, color shadow register number, and then, register. This table makes it handy to make quick reference to whatever mode you want.

The second chapter is a complete overview of a display list, explaining everything needed to set up your own. It also comes with a lot of programs showing you exactly how a display is set up.

The third chapter was one that I really enjoyed: It gave one of the best explanations of Character Set Graphics that I have ever read. It starts off by explaining what a simple character is, and goes on to explain how to make multicolor character graphics, and how to rotate character sets and animation. This one section is, in itself, worth the cost of the book.

In the fourth chapter, assembly language is written about. It is called, "Assembly Language Applied To Game Design," and the title defines exactly what it is. In this chapter, the goal of the authors is to teach the fundamentals of Assembly language programming by comparing it to the similar code written in BASIC. Rather than teaching all the aspects of the language, they concentrate only on the operations needed for simple game graphics. First, a listing of the simple game "Breakout" is given in BASIC for the user to type in. After studying it, he can type in the assembler listing of the game.

This type of instruction teaches the BASIC user the difference between the two languages, and should gradually ease the user into the world of assembly work. Although the assembly listing is much longer than the BASIC listing, it is worth it in the long run, because assembler code is always much faster.

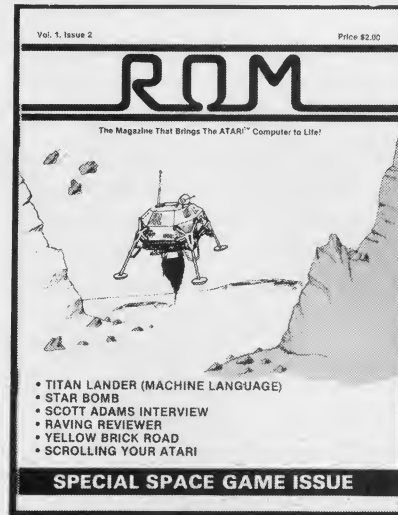
The fifth chapter is one regarding a much explained topic, "Player/Missile Graphics." Since there has been so much written on the subject you wouldn't think that anything new could be said; However, this is where you would be wrong. It explains things like Dynamics of objects in motion, something I have never seen discussed anywhere except here. It gives a brief example of a ship that flies around and fires missiles. This program, which was written in BASIC, is quite fast because of the use of an assembly language subroutine. Next, is a BASIC program that has two ships that fire at each other, and when one ship is hit, the collision register is activated.

The assembler listing of the two ship shootout game is given, showing the great speed of machine language. Finally, to finish off this chapter, is a simple

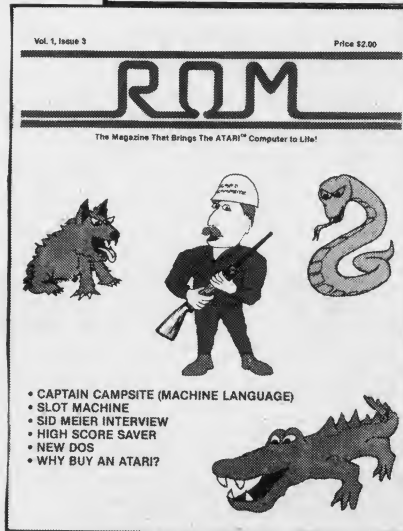
# LIMITED BACK ISSUES AVAILABLE!!



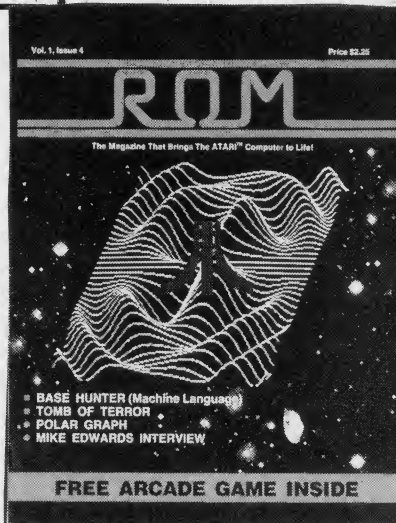
Original  
Number  
One  
\$5.00 U.S.



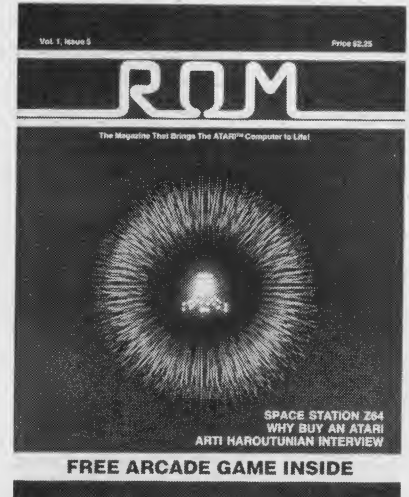
\$4.00 U.S.



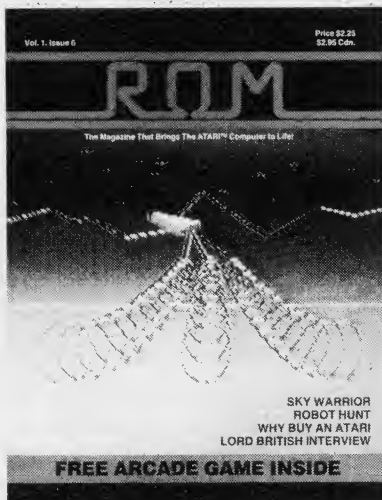
\$4.00 U.S.



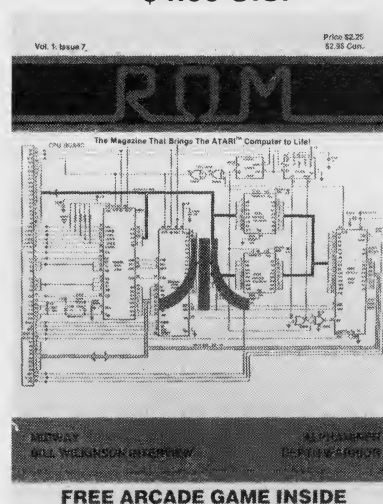
\$4.00 U.S.



\$4.00 U.S.



\$4.00 U.S.



\$4.00 U.S.



\$4.00 U.S.

but good Player/Missile Editor. This section alone is nearly one hundred pages in total, thus giving you a great amount of information.

Chapter six is called, "Vertical Blank & Display List Interrupts," and tries to explain how to use them. These interrupts are a powerful aid to the game programmer who can use them to smooth animation, to enable players to be re-used in the bottom portion of the frame, to allow character sets, and to enable color registers to be changed mid-screen, and, of course, to do much more. This book gives a very thorough explanation of Kernels in a Display List Interrupt routine to control graphics information on a line-by-line basis for the entire screen. A couple of assembly language programs are then listed, including one on using DLI's to create animation.

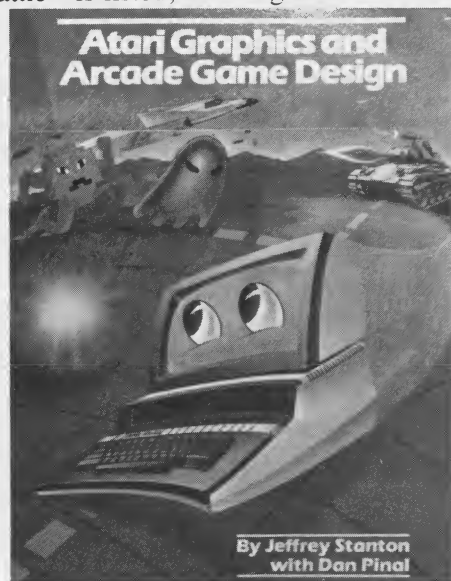
Chapter seven, "Games that Scroll," is one that hasn't been explained very well until now. First, a brief example of coarse vertical and horizontal scrolling is given in BASIC, showing how easy it can be done. Fine scrolling is then explained, and is followed by an eight way scrolling assembler listing. To finish off this chapter there is a complete listing of "Strike Force-A Scrolling Game." This game, which features fine horizontal scrolling, has a ship fly over missile bases and lasers. Your ship can drop bombs, or fire lasers to destroy the alien bases or ships. Since the game has good documentation throughout the listing, it makes it easy for the user to become familiar with the techniques used.

Raster Graphics & Sound is explained in chapter eight. Raster graphics is a term we very rarely use in connection with the Atari computer system. It is a term that describes how individual pixels are mapped on a high-resolution screen. The technique is about the only one possible on computers such as the Apple II and the IBM PC. Atari programmers like to use easier and more colorful techniques like character graphics and player/missile animation, but there are certainly a number of valid reasons for animating with raster graphics. The two best reasons are that Graphics mode 8 screens have the highest resolution, and that very large shapes can be smoothly animated. This is a subject that isn't explained in many articles, and is a very good technique if a person wants to take the time to use it.

This type of graphics is used in Ultima III because it is easier to convert programs from an Apple to an Atari, since they both use the same method of producing graphics. A good example is given in an assembler listing, showing a blimp that can be moved around the screen with a joystick.

Chapter nine, which is called "Advanced Arcade Techniques," is one that explains methods, or algorithms, of creating maze games. While most people do not think of games like Donkey Kong and Apple

Panic as maze games, they, too, require a set of movement rules to keep the player confined to floors and ladders. In this chapter, they discuss how to create computer controlled characters to move with some logical movement. Then, at the end of the section, the assembler listing to the game "Alphabet Maze" is given. Next, the authors show how to design a tank game from scratch. Each player controls his or her own tank which can fire in any of eight directions while travelling in the opposite one. Finally, at the end of this chapter, a complete listing of the game "Tank Battle" is listed, showing all of the comments beside it.



The final chapter, "Game Design Theory" is an overview of game design, and of what type of game should be successful. It says things such as, "There is no sure-fire way to predict whether a game will be successful, but there are certain attributes that contribute to success." It gives examples of successful arcade games, and reasons for their success. It explains why games like Vanguard, Pole Position, Joust, Pacman, Donkey Kong, and Frogger have all succeeded. In short, they say you should plan out your game completely before you start programming, or the disorganization will show up later in the finished product. If you make a game quickly and hope it will be a success, you will probably be disappointed because only a lot of time and hard work will make a game a success.

This book, "Atari Graphics and Arcade Design" is 478 pages long, and retails for the low price of \$16.95 US, making this one of the best books for the Atari to come along. The fact that all of the programs in the book are also available on disk, saves a lot of typing mistakes. This book is available from:

Arrays, Inc./The Book Division  
11223 South Hindrey Ave.  
Los Angeles, California 90045

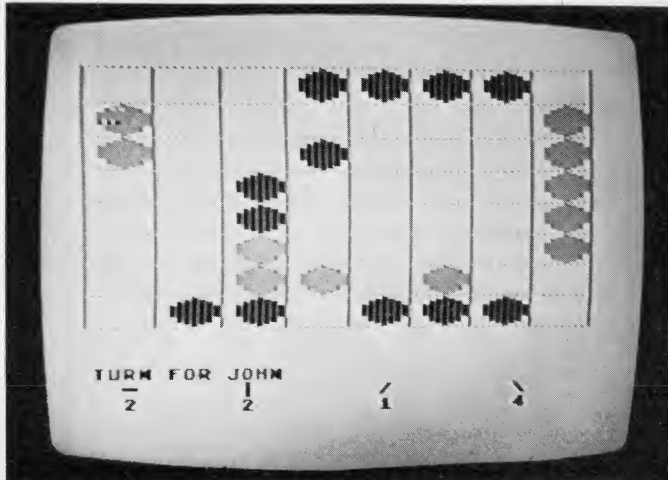


# Lines Of Action

by Sol Guber

There are many action board games in which the possible moves are very dependent on the previous moves. The most famous of these is OTHELLO, where each move will change the configuration of the board, and a whole new strategy is needed. LINES OF ACTION by Claude Soucie is related to OTHELLO, but it is both more complicated, and simpler to play. Figure 1 shows the board at the beginning of play. The object of the game is to have your men touching one another vertically, horizontally, and diagonally. You can also take the other person, but if he has only one man left, he automatically wins (one man always touches itself). Also, the fewer men there are on the board, the easier it is to win.

The play of the game is simple. Each person, in turn, moves any man to another spot on the board. The direction that a man can move, and the number of



squares that he can move, depends on his position on the board, and on the position of all the other men. A man can move horizontally, vertically, or diagonally the same number of squares as there are men in that direction. Let me explain this further. Looking at Figure 1 again, observe the WHITE man at B2. He can move horizontally, vertically, diagonally right, and diagonally left. In his horizontal row, there are six men. Thus, he can move (in theory) six squares to the left, or six squares to the right in a horizontal manner. There are two men in the vertical column, so he can move two squares up, or two squares down. There are two men in the diagonal left column, so he can move two squares in that direction. There are two men in the diagonal right, so he can move two squares in that direction. During play, you can see all the possible moves of a man, by moving the cursor to that man, and pressing the trigger. These are all possible moves. There are limits to the moves that can be made in the actual game. You cannot move off the board. You

cannot move and land on your own man. You can jump over your own men, but you cannot jump over an opponent's man. You can land on your opponent, and he is removed from the board. Otherwise, the moves can be made in any manner.

Let me go into the way that the game is played. When the game is started, there is a question regarding instructions. A "Y" or "N" answer is expected. The computer asks for the names of the two players. The game can be played at two levels of difficulty. At the bottom of the screen, all the possible moves of any man can be determined by the Computer, and shown. If you do not want this to be done, answer the question with an "N." When it is a person's turn, his name will be shown at the bottom of the screen. White uses Joystick 1, and BLACK uses Joystick 2, to move a flashing cursor on the screen.

If the direction option is on, the cursor is moved to the man you wish to examine, and the trigger is pressed. The bottom display changes, and a marker is put on that man. The cursor is then moved to either the position to which you want to move that man, or to another man of the same color. Then the trigger is pressed. If it is to a man of the same color, the old marker is moved, and the information on the screen is updated. If it is not to a man of the same color, the move is checked to see if it is a legal one. If it is not, the move must be taken over. If it is a legal move, the man is moved to that square and removed from the old square. The board is checked to see if anyone has won, and then it is the other person's move. The game continues until someone wins.

The game has a large amount of complexity to it, yet it is very simple to learn. To help the beginner, there is the option to show on the bottom of the screen all the possible moves. For the advanced player, this is too distracting, and is not needed. The strategy of the game involves all the different men that can be moved. The decision whether to jump an opponent and remove him from the board is tempered with the idea that the fewer men on the board, the easier it is to win. Since the object of the game is the same for both players, each play must be both offensive and defensive at the same time.

To go through the working of the strategy, let's review the play of a game. In Figure 1, the men are set up as they would be at the start. The WHITE player is conservative, and plans his moves carefully. BLACK plays a more reckless game, and keeps trying to remove WHITE's men. The first move by WHITE is from A4 to C4. The man on A4 can move six squares in a horizontal position, and two squares in

each of the vertical, the left diagonal, and the right diagonal positions. He has moved two squares downward in a vertical manner. On his next move, if he so wishes, he can move back to his original square. The BLACK man on C1 can move six squares vertically, three squares horizontally, and two squares either left diagonal, or right diagonal. He wishes to move three squares horizontally to the right, and lands on the WHITE man, removing him from the board in Figure 3. Again, this is a combination of plays. To be too aggressive, and to remove your opponent's men, will make it easier for him to win.

The play for the next few moves is rather straightforward. WHITE's next move is just a vertical one down. BLACK then moves horizontally from H3 to E3. White then moves vertically up from E8 to E5 in Figure 6. There are many more plays, and these are detailed in Table I, which shows the whole game.

Let me skip ahead several moves to Figure 7, which corresponds to move 10 for WHITE. By Figure 7, Black has a decided advantage. He has ten of his eleven men touching, and all he needs is to move the man at H7 to the proper position, and he will win. White must prevent Black from bringing all his men together. To do this, White moves vertically up two to block BLACK's diagonal left move. If Figure 9, Black moves diagonally down from H7 to G8, so that on his next move, he can move diagonally left 3 up, and win the game. In Figure 10, White moves from G6 to F7 to block this move, since BLACK cannot jump over his opponent's men. Black moves from G8 to G6, to try some other method of winning the game.

WHITE now has a chance to make some counterattacks. In Figure 12, move 12, White goes from B1 to D1. BLACK moves from G6 to G4, to win by a move to E2. WHITE moves from D6 to D3, on a vertically up move, removing the BLACK man on D3. White has now split up BLACK into two parts, and now BLACK has lost the offensive. In Figure 15, BLACK has moved from G4 to E2, to try to find some method of bringing his two groups together. WHITE moves from G1 to G2, to try to take the man at E2. BLACK moves his man away from D2 to B2.

WHITE has changed the flow of the game. White now moves from G2 to E4, taking the BLACK man there. White is now trying to consolidate his men, in Figure 18. BLACK now moves from C2 to D2, and WHITE moves from D1 to E1 in move 16. In Figure 19, BLACK moves from A6 to C6, to try to get the man at E4. White counters by moving from F7 to C4, removing the BLACK man there. BLACK then moves from C6 to E4, taking WHITE. WHITE moves from F4 to F2. BLACK moves from E4 to F3, to try to take the man at F6, but WHITE does it first in Figure 26. BLACK moves from E5 to C3, taking WHITE there, and leaves himself open for WHITE's

E1 to E3 move for the win.

I have gone through a game in a great deal of detail to show the flavor of it. It is very exciting, and contains portions of OTHELLO, checkers, and chess. The interesting part of the game is how each move changes the board for the other men. The major flaw is in Figure 27, where Black is so eager to move to C3 that he removes one man from that column, allowing WHITE a two move vertically, which removed the man. If the man were still at E5, then only a three vertical move would be allowed, and the man at E3 would have been safe.

I am willing to make cassette copies of this program for those that would like them. Please send a self addressed, stamped cassette mailer to me, plus \$3 to cover postage and handling.

Send the cassette to:

Sol Guber  
717 Elkington  
St. Louis, MO 63132  
USA

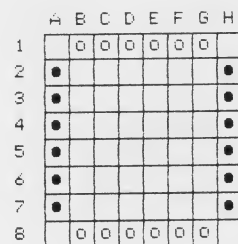


Figure 1

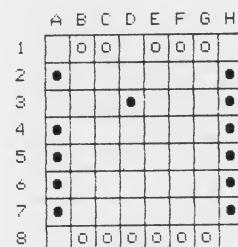


Figure 3

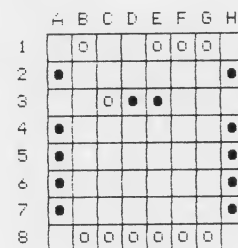


Figure 5

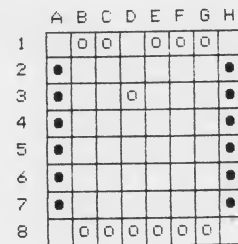


Figure 2

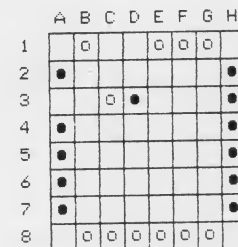


Figure 4

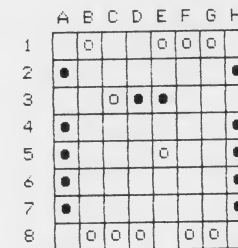


Figure 6

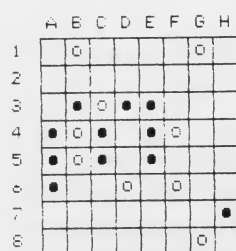


Figure 7

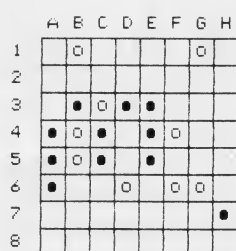


Figure 8

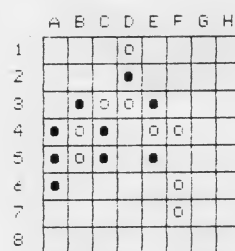


Figure 19

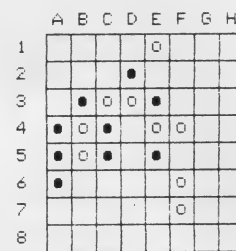


Figure 20

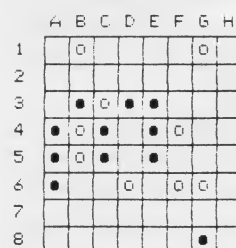


Figure 9

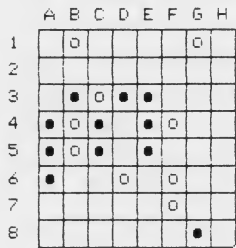


Figure 10

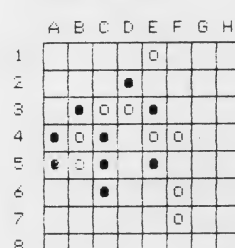


Figure 21

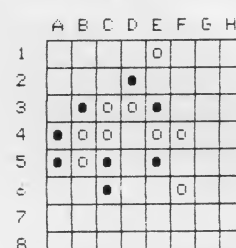


Figure 22

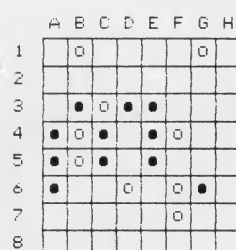


Figure 11

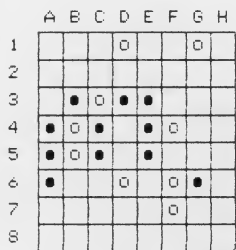


Figure 12

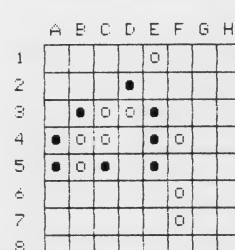


Figure 23

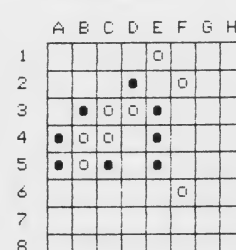


Figure 24

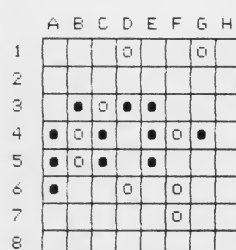


Figure 13

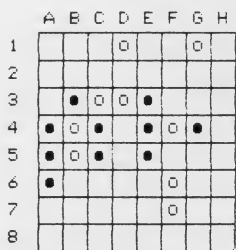


Figure 14

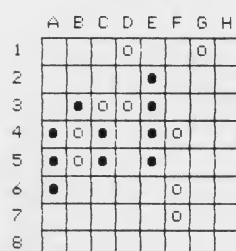


Figure 15

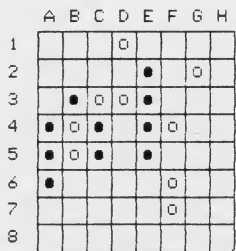


Figure 16

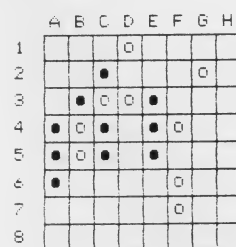


Figure 17

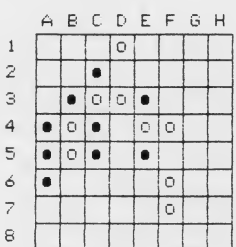


Figure 18

TABLE ONE

	White	Black
1.	D1-D3	A3-D3
2.	C1-C3	H3-E3
3.	E8-E5	H2-E5
4.	E1-B4	H4-E4
5.	B8-B5	A2-C4
6.	C8-C5	A4-B3
7.	D8-D6	A7-A4
8.	F8-F6	H6-F4
9.	F1-F4	H5-C5
10.	G8-G6	H7-G8
11.	G6-F7	G8-G6
12.	B1-D1	G6-G4
13.	D6-D3	G4-E2
14.	G1-G2	E2-C2
15.	G2-E4	C2-D2
16.	D1-E1	A6-C6
17.	F7-C4	C6-E4
18.	F4-F2	E4-F3
19.	F6-C3	E5-C3
20.	E1-E3	

Underlined man is taken.



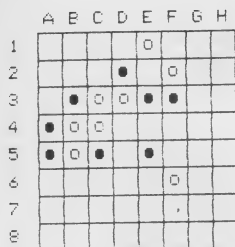


Figure 25

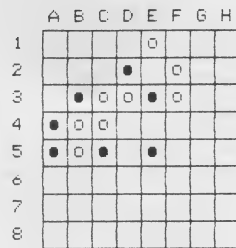


Figure 26

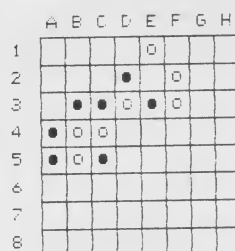


Figure 27

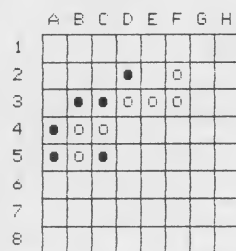


Figure 28

```

1 AGAIN=0:REM LINE OF ACTION BY SOL
  GUBER.
2 C1=1:C0=0:C2=2:C3=C2+C1:C4=C2+C2:C5=
5:C6=C4+C2:C7=C5+C2:C8=C6+C2:C9=C8+C1:
C10=C2+C8:C15=C10+C5:C14=C15-C1:C17=17
4 DIM R$(120),S$(120),PAT$(8):R$(C1)=C
HR$(C0):R$(120)=CHR$(C0):R$(C2)=R$:C12
=12:C11=C12-C1
5 DIM PLAYERS(20),AS$(C10),XDIF(C15),YD
IF(C15):PLAYERS(C1)="" :PLAYERS(20)=""
":PLAYERS(C2)=PLAYERS
6 IF AGAIN<>C0 THEN GO TO C10
7 GOSUB 4000:GOSUB 6000
8 TRAP C8:? "KDO YOU NEED INSTRUCTIONS
":INPUT AS$:IF AS$(1,1)="Y" THEN GOSUB
3000
10 GOSUB 3500:GRAPHICS C0:DIM MOVE(C4)
,MOVT(C7),KN(C2):GOSUB 1000
15 DX=C12:DY=C3:DPX=DX:DPY=DY:PT=C2:PO
KE 752,C1:GOSUB 200
18 KN(C0)=12:KN(C1)=12:MOVE(C0)=C0
30 FOR I=C0 TO C1:FLG=C0:POKE 77,C0:VL
=(I=C0)*C5+(I=C1)*C10
35 ? "TURN FOR ";PLAYERS(C10*I+C1,C10
*I+C10)
36 IF CFLAG=C1 THEN 45
38 IF MOVE(C0)=C0 THEN ? "
":GOTO 45
40 ? " -
":? " ";MOVE(C0),MOVE(C1),MOVE(C2),MOV
E(C3)
45 COLOR C3:PLOT DPX+C3,DPY+C9:PLOT DP
X+C4,DPY+C9:PLOT DPX+C3,DPY+C10:PLOT D
PX+C4,DPY+C10
50 S=STICK(I):TR=STRIG(I):IF JOY=C1 TH
EN S=STICK(C0):TR=STRIG(C0)
53 IF TR=C0 AND FLG<>C1 THEN FLG=C1:KN
T=(DPX+C6)/C9+C10*(DPY-C3)/C17:GOSUB 3
00:GOTO 35
55 IF TR=C0 AND FLG=C1 THEN 115
58 IF S=C15 THEN LOCATE DPX,DPY,Z:POKE
707,PEEK(704):Q=1^1^1:POKE 707,Z:Q=1^
1^1:GOTO 50
60 DX=DPX+XDIF(S)*C9:DY=DPY+YDIF(S)*C1
7
63 IF DX<C3 OR RX>67 THEN 50
65 IF DY<C3 OR DY>122 THEN 50
85 LOCATE DPX+C6,DPY+C9,Z:COLOR Z:PLOT
DPX+C3,DPY+C9:PLOT DPX+C4,DPY+C9:PLOT
DPX+C3,DPY+C10:PLOT DPX+C4,DPY+C10
88 DPX=DX:DPY=DY
90 FOR Q=C1 TO 75:NEXT Q:GOTO 45
115 GOSUB 550:GOTO 320
120 SOUND C0,C0,C0,C0
135 ? "KYOUR KRN MUST BE MOVED
":GOSUB 520:FOR WAIT=C1 TO 250:
NEXT WAIT
140 FLG=C0:RETURN
190 NEXT I:GOTO 30
200 IF ASC(R$(PT))=C0 THEN FOR Q=C0 TO

```

```

C4:MOVE(Q)=C0:NEXT Q:RETURN
203 TPT=INT(PT/C10):UPT=PT-TPT*C10
205 COUNT=C0:K3=TPT*10+C1:K4=TPT*C10+8
:K5=C1:GOSUB 250:MOVE(C0)=COUNT
210 COUNT=C0:K3=UPT:K4=UPT+70:K5=C10:G
OSUB 250:MOVE(C1)=COUNT
215 T1PT=TPT:U1PT=UPT:IF TPT+UPT>C9 TH
EN T1PT=C8-UPT:U1PT=C8-TPT
220 COUNT=C0:K3=PT-T1PT*C9:K4=PT+(U1PT
-C1)*C9:K5=C9:GOSUB 250:MOVE(C2)=COUNT
225 T1PT=TPT:U1=T=UPT:IF UPT>TPT THEN
U1PT=TPT+C1:T1PT=UPT-C1
230 COUNT=C0:K3=PT-(U1PT-C1)*C11:K4=PT
+(C7-T1PT)*C11:K5=C11:GOSUB 250:MOVE(C
3)=COUNT
235 RETURN
250 FOR K2=K3 TO K4 STEP K5:IF ASC(R$(
K2))<>C0 THEN COUNT=C(UNT+C1
260 NEXT K2
270 RETURN
300 GOSUB 500
302 IF I=C0 ANDOASC(R$(KNT))<>C5 THEN
135
303 IF I=C1 AND ASC(R$(KNT))<>C10 THEN
135
305 PT=(DPX+C6)/C9+C10*(DPY-C3)/C17:GO
TO 200
320 K2=(DPX+C6)/C9+C10*(DPY-C3)/C17:IF
ASC(R$(K2))=VL THEN FLG=C0:GOSUB 600:
KNT=K2:GOSUB 305:GOSUB 503:GOTO 35
322 IF ABS(K2-KNT)=MOVE(C0) THEN 340
324 IF ABS(K2-KNT)=MOVE(C1)*C10 THEN 3
40
326 IF ABS(K2-KNT)=MOVE(C2)*C9 THEN 34
0
328 IF ABS(K2-KNT)=MOVE(C3)*C11 THEN 3
40
338 ? "KILLEGAL MOVE - TRY AGAIN
":GOSUB 520:GOSUB 600:FLG=C0:FO
R WAIT=C1 TO 250:NEXT WAIT:GOTO 35
340 VL=(I=C0)*5+(I=C1)*10:K4=K2-KNT
342 IF K4/C11=INT(K4/C11) THEN K5=11:G
OTO 350
344 IF K4/C10=INT(K4/C10) THEN K5=C10:
GOTO 350
346 IF K4/C9=INT(K4/C9) THEN K5=C9:GOT
O 350
348 K5=C1
350 IF K2<KNT THEN K5=-K5
353 FOR Q=KNT TK K2-K5 STEP K5:IF ASC(
353 FOR Q=KNT TO K2-K5 STEP K5:IF ASC(
R$(Q))<>VL AND ASC(R$(Q))<>C0 THEN 390
355 NEXT Q
358 IF ASC(R$(K2))=VL THEN ? "KYOU
CANNOT LAND ON YOUR OWN KRN":GOSUB
520:FLG=C0:GOSUB 600:GOTO 31
360 IF ASC(R$(K2))<>C0 THEN KN((I=C0))
=KN((I=C0))-C1
363 COLOR (I=C0)*C7+(I=C1)*C5:P=INT(K2
/C10):X=(K2-P*C10)*C9-C6:Y=P*C17+C3:GO
SUB 2300
365 COLOR C0:P=INT(KNT/C10):X=(KNT-P*C
10)*C9-C6:Y=P*C17+C3:GOSUB 2300
368 R$(KNT,KNT)=CHR$(C0):R$(K2,K2)=CHR
$(VL)
370 M1=C0:VL=C5:GOSUB 400:M1=C1:VL=C10
:GOSUB 400:MOVE(C0)=C0:GOTO 195
390 ? "KYOU CANNOT JUMP OVER YOUR OPP
ONENT":GOSUB 520:GOSUB 600:FLG=C0:FO
R WAIT=C1 TO 250:NEXT WAIT:GOTO 35
400 J=C1:S=S:R=COUNT=C1
403 IF ASC(S$(J))<>VL THEN J=J+1:GOTO
403
405 K=C0
412 P1=J+MOVT(K):IF P1<C1 THEN 450
415 K2=ASC(S$(P1)):IF K2<>VL THEN 450
420 S$(J,J)=CHR$(VL+K*16):COUNT=COUNT+
C1:J=P1:GOTO 405
450 K=K+C1:IF K<C8 THEN 412
455 S$(J,J)=CHR$(VL+112):IF COUNT>=KN(
M1) THEN GOTO 700+100*M1
460 FOR K=C0 TO C7:P1=J+MOVT(K):IF P1<
C1 THEN 480
463 K2=ASC(S$(P1)):IF K2=C0 THEN 480
465 IF K2=VL+112 THEN 480
468 IF K2<C11 THEN 480
470 K=INT(K2/16)+1:J=P1:GOTO 412
480 NEXT K:RETURN
500 SOUND 0,75,10,12:FOR I1=1 TO 100:N
EXT I1:SOUND 0,0,0,0
503 IF ASC(R$(KNT))<>VL THEN FLG=C0:RE
TURN
510 COLOR 1:PLOT DPX+C4,DPY+C3:PLOT DP
X+C5,DPY+C3:RETURN

```

```

520 SOUND 0,75,12,12:FOR I1=1 TO 100:M
EXT I1:SOUND 0,0,0,0:RETURN
550 SOUND 0,125,10,12:FOR I1=1 TO 100:
NEXT I1:SOUND 0,0,0,0:RETURN
580 SOUND 0,125,12,12:FOR I1=1 TO 100:
NEXT I1:SOUND 0,0,0,0:RETURN
600 P=INT(KNT/C10):X=(KNT-P*C10)*C9-C6
:Y=P*C17+C3
603 LOCATE X+C6,Y+C9,Z:COLOR Z
610 PLOT X+C4,Y+C3:PLOT X+C5,Y+C3:RETU
RN
700 ? "K";PLAYERS(C1,C10):? "IS THE WI
NNER":FOR I1=0 TO 255 STEP 0.25:POKE 7
08,I1:NEXT I1
710 GOTO 805
800 ? "K";PLAYERS(11,20):? "IS THE WIN
NER":FOR I1=0 TO 255 STEP 0.5:POKE 708
,I1:NEXT I1
805 FOR WAIT=C1 TO 300:NEXT WAIT:POKE
764,255
810 TRAP 810:? "KPLAY AGAIN";:INPUT PL
AYERS:IF PLAYERS(1,1)="Y" THEN CLR :AG
AIN=1:TRAP 40000:GOTO 2
820 TRAP 40000:GRAPHICS 0:END
1000 GOSUB 4000:GOSUB 4045
1005 RESTORE 1020
1010 FOR I=C0 TO C7:READ A:MOUT(I)=A:M
EXT I
1020 DATA -11,-1,9,10,11,1,-9,-10
1200 ? "KWHITE PLAYER, PLEASE ENTER YO
UR NAME":INPUT A$:IF LEN(A$)=C0 THEN 1
200
1205 PLAYERS(1,LEN(A$))=A$:I=C0
1210 ? "KBLACK PLAYER, PLEASE ENTER YO
UR NAME":INPUT A$:IF LEN(A$)=C0 THEN 1
200
1215 PLAYERS(11,10+LEN(A$))=A$:I=C1
1220 RESTORE 1225
1225 DATA 1,1,1,-1,1,0,0,0,-1,1,-1,-1,
-1,0,0,0,0,1,0,-1,0,0
1230 FOR Q=C5 TO 15:READ X,Y:XDIFF(Q)=X
:YDIFF(Q)=Y:NEXT Q
1250 ? "KADVANCED PLAY (Y/N)"::INPUT A
$:IF LEN(A$)=C0 THEN RETURN
1260 IF A$(1,1)="Y" THEN CFLAG=C1
1300 RETURN
2300 PLOT X+C2,Y+C5:DRAWTO X+C2,Y+C11
2310 PLOT X+C3,Y+C3:DRAWTO X+C3,Y+C14
2320 PLOT X+C4,Y+C2:DRAWTO X+C4,Y+C15
2340 PLOT X+C5,Y+C1:DRAWTO X+C5,Y+16
2350 PLOT X+C6,Y+C2:DRAWTO X+C6,Y+C15
2360 PLOT X+C7,Y+C3:DRAWTO X+C7,Y+C14
2370 PLOT X+C8,Y+C5:DRAWTO X+C8,Y+C11
2380 RETURN
3000 TRAP 40000:GRAPHICS C0:POKE 709,C
0:POKE 710,190:POKE 712,190:POKE 752,C
1
3002 ? "
:?"
3005 ? "THIS IS A VERY INTRICATE GAME.
THE OBJECT IS TO POSITION ALL YOUR
PIECES"
3010 ? "IN SUCH A MANNER THAT THEY TOU
CH ONE ANOTHER, REGARDLESS OF THE ACT
UAL"
3015 ? "FORMATION OR THE NUMBER OF MEN
LEFT ON THE BOARD."
3020 ? "THINK OF YOUR MEN AS PIECES OF
A CHAIN. YOUR JOB IS TO CONNECT
ALL"
3025 ? "THE LINKS. THE LENGTH OR SHAP
E OF THE CHAIN IS IRRELEVANT. IN F
ACT,"
3028 ? "THE MORE LINKS YOU HAVE, THE M
ORE DIFFICULT YOUR JOB BECOMES."
3029 ? "FOR EXAMPLE, IF YOU ONLY HAVE
ONE PIECE LEFT, YOU AUTOMATICALLY
WIN!"
3030 ? "YOUR MOVEMENTS ARE GOVERNED BY
CERTAIN RULES."
3050 POKE 764,255:POSITION 6,21:? "PR
ESS ANY KEY TO CONTINUE"
3060 IF PEEK(764)=255 THEN 3060
3065 ? "K
:?"
3070 ? "ALTHOUGH YOU MAY MOVE VERTICAL
LY, HORIZONTALLY, DIAGONALLY, FORM
ARD"
3080 ? "AND IN REVERSE, YOUR LIMITS AR
E AS FOLLOWS:"
3085 ? " 1. THE NUMBER OF SQUARES YOU
R PIECE CAN MOVE IS DETERMINED BY ITS
PRESENT"
3090 ? "POSITION, AS WELL AS BY HOW MA
NY OTHERMEN ARE IN THE SAME ROW OR COL

```

```

UMN."
3092 ? "AFTER YOU INDICATE WHICH PIECE
YOU INTEND TO PLAY WITH, THE COMPU
TER WILL"
3094 ? "DISPLAY ITS VARIOUS AVAILABLE
MOVES."
3095 ? " 2. YOU MAY JUMP OVER YOUR ME
N, BUT NOT OVER YOUR OPPONENT'S."
3100 ? " 3. YOU ARE ALLOWED TO LAND O
N YOUR OPPONENT'S PIECE (NOT YOURS).
THIS ELIMINATES THAT PIECE";
3105 ? " FROM THE GAME."
3110 POKE 764,255:POSITION 6,21:? "PR
ESS ANY KEY TO CONTINUE"
3115 IF PEEK(764)=255 THEN 3115
3120 ? "K
:?"
3125 ? "IN THE ADVANCED-PLAY MODE, THE
ALLOW-"
3128 ? "ABLE MOVES FOR THE VARIOUS PIE
CES WILL NOT BE SHOWN."
3130 ? "POSITION THE CURSOR ABOVE THE
MAN YOU WISH TO MOVE AND PRESS THE
FIRE-"
3135 ? "BUTTON. THEN MOVE THE CURSOR
TO THE SQUARE WHICH YOU WANT TO LAND
ON AND"
3140 ? "PRESS THE FIREBUTTON AGAIN. I
F YOU CHANGE YOUR MIND, SIMPLY LAND
ON"
3142 ? "THE SAME MAN YOU STARTED OUT W
ITH. YOU MAY THEN MAKE A DIFFERENT
MOVE."
3144 ? "IF YOUR MOVE IS ILLEGAL, THE C
OMPUTER WILL GIVE YOU THE REASON WHY."
3145 ? "IN A TWO-JOYSTICK GAME, WHITE
ALWAYS USES PORT #1; BLACK USES PORT
#2."
3148 ? "THINK AHEAD BEFORE MOVING. GOO
D LUCK!"
3150 POKE 764,255:POSITION 6,21:? "PR
ESS ANY KEY TO CONTINUE"
3160 IF PEEK(764)=255 THEN 3160
3400 POKE 752,C0:RETURN
3500 TRAP 3500:? "K":POSITION 2,12:? "
1 OR 2 JOYSTICKS";:INPUT JOY:IF JOY<C1
OR JOY>C2 THEN 3500
3510 TRAP 40000:RETURN
4000 RESTORE 4005:GRAPHICS 8:FOR I=704
TO 712:READ A:POKE I,A:NEXT I
4005 DATA 12, 76,86,68,130,0,190,14,19
0
4010 Z=PEEK(560)+256*PEEK(561):POKE 55
9,0:POKE Z+166,143:POKE 513,6:POKE 512
,0
4015 RESTORE 4030
4020 FOR I=1536 TO 1546:READ A:POKE I,
A:NEXT I
4030 DATA 72,169,0,141,10,212,141,27,2
08,104,64
4040 POKE 54286,192:POKE 559,34
4042 POKE 87,C10:POKE 623,128:RETURN
4045 COLOR C4
4050 FOR I=C3 TO 139 STEP C17:PLOT C3,
I:DRAWTO 75,I:NEXT I
4060 FOR I=C3 TO 75 STEP C9:PLOT I,C3:
DRAWTO I,139:NEXT I
4070 Y=C3:COLOR C7
4073 FOR X=C12 TO 60 STEP C9:GOSUB 230
0:NEXT X
4075 Y=122:FOR X=C12 TO 60 STEP C9:GOS
UB 2300:NEXT X
4080 COLOR C5:X=C3:FOR Y=20 TO 105 STE
P C17:GOSUB 2300:NEXT Y
4090 X=66:FOR Y=20 TO 105 STEP C17:GOS
UB 2300:NEXT Y
4100 FOR X=C2 TO C7:R$(X,X)=CHR$(C5):N
EXT X
4110 FOR X=72 TO 77:R$(X,X)=CHR$(C5):N
EXT X
4120 FOR X=C11 TO 61 STEP 10:R$(X,X)=C
HR$(C10):NEXT X
4130 FOR X=18 TO 68 STEP 10:R$(X,X)=CH
R$(C10):NEXT X
4300 RETURN
6000 RESTORE 6100
6002 COLOR C4:FOR I=C3 TO 139 STEP 17:
PLOT C3,I:DRAWTO 75,I:NEXT I
6004 FOR I=C3 TO 75 STEP 9:PLOT I,3:DR
AWTO I,139:NEXT I
6010 GOSUB 6500
6030 FOR P=C1 TO 40:X=PEEK(704)
6040 FOR I=704 TO 711:POKE I,PEEK(I+1)
:NEXT I:POKE 711,X:NEXT P
6045 SOUND C0,C0,C0,C0

```

```

6050 RETURN
6100 DATA 22,44,3,33,41,3,44,46,3,55,3
7,3,66,51,3,35,47,7,36,38,7
6110 DATA 6,33,1,15,35,1,24,52,1,42,47
,1,51,46,1,62,51,4,63,47,4,64,44,4
6115 DATA 71,39,5,72,53,5,73,34,5,74,3
7,5,75,50,5
6500 FOR P=1 TO 20:READ Z,K,K2:COLOR K
6520 TR=57344+C8*K+C1:FOR J=C0 TO C7:P
AT(J)=PEEK(TR+J):NEXT J
6600 X4=INT(Z/C10):Y4=Z-X4*C10:X4=X4*1
7+C6:Y4=Y4*C9-C6
6610 FOR J=C0 TO C6:X2=PAT(J):SOUND C0
,K2,C10,C10
6620 FOR J2=C0 TO C8
6630 IF K2/C2<>INT(K2/C2) THEN PLOT Y4
+C8-J2,X4+J*C2:PLOT Y4+C8-J2,X4+J*C2+C
1
6640 K2=INT(K2/C2):NEXT J2
6660 NEXT J:NEXT P:RETURN

```



# PANZER

\$19<sup>95</sup>

SENTINEL SOFTWARE  
P.O. BOX 11701  
FRESNO, CA 93774



24  
HOUR  
SHIPPING

## ATARI™ USERS

WE SPECIALIZE IN BACKUP HARDWARE AND SOFTWARE

"We are the Backup Experts"

24  
HOUR  
SHIPPING



### NEW PRODUCT The DOWNLOADER For The ATARI 835 Modem™

At last, a program that will allow you to download Binary and Basic files with the new 835 Modem, no interface needed. You can save these files to disk, printer or cassette. But **Best Of All** you will be able to download games from bulletin boards with our software and the 835 Modem.

**\$34<sup>95</sup>**

+ 2.50 Shipping

### THE BOOK WITH SOFTWARE

Software protection and code cracking techniques. **MASTER CODE CRACKER REVEALS ALL**. In this book you will find out how the software is protected and ways to protect your software. Copy guarding will be covered in detail on disks, cartridges and tapes and hardware tricks. You will also receive a disk with many programs and examples.  
**BOOK WITH SOFTWARE ONLY**

**\$24<sup>95</sup>**

+ 2.50 Shipping

ATARI™ & 810 ARE TRADEMARKS OF ATARI

### 835 & 1030 MODEM BULLETIN BOARD

This BBS Bulletin Board system will run on any ATARI Home Computer including the XL. No costly interface needed. All you will need is an 835 or 1030 Modem and any disk drive (printer optional). Auto Answer feature will allow you to leave the BBS running unattended. This BBS has over 25 functions including: XMODEM Upload/Downloading, User Passwords, Full Function Message Base plus many more features.

This package comes on a double sided disk, full documentation included plus a fully assembled and tested ring detector. Nothing else will be needed. BBS software and ring detector:

All For Only **\$74<sup>95</sup>** + 2.50 Shipping

### THE SUPER TRANSLATOR

For The ATARI 800XL  
A MUST FOR THE NEW 800XL

Many programs written for the old ATARI 800 Computer will not run on the new 800XL. With the **SUPER TRANSLATOR** you will be able to run 40% more software. The **SUPER TRANSLATOR** comes on disk as a file. Order now and receive **FREE** a DOS Patch File. This will relocate ATARI DOS in the unused 16K of memory. You will now have the entire DOS in RAM. No need to access the disk when you call DOS.

All For **\$14<sup>95</sup>** + 2.50 Shipping

### The HACKER'S TREASURE CHEST

On Disk

18 Utility Programs on disk. These programs are designed to aid you in copying software for your backup collection. You will be able to copy disks, cartridges and cassettes. Any one program is worth the price of all 18. It has taken us over one year to put together this fine collection on the **Hacker's Treasure Chest** disk. Some of the programs you will receive are: **Cartridge Copy, Boottape Maker, Tape to Disk, Sector Copy, The Unprotector, Sector Disassembler, Bad Sector Finder, Modem Program**... plus more. All of these programs plus 10 more on this disk. You will also receive a menu that will run basic and binary files just by typing the number of the program. Any file on any disk will load automatically from this menu. **ALL FOR ONLY**

**\$24<sup>95</sup>**

+ 2.50 Shipping

### The TRANSFER PACK FOR BACKING UP AND TRANS- FERRING YOUR SOFTWARE

- 1) Disk file to tape
- 2) Boot tape to disk file
- 3) Tape to tape (multi & single stage)

VERY POWERFUL and low priced. Programs are in machine language and user friendly. **ALL 3 FOR ONLY**

**\$24<sup>95</sup>**

+ 2.50 Shipping

PHONE ORDERS  
(516) 467-1866  
PRODUCT INFO  
(516) 588-6019

## GARDNER COMPUTING COMPANY

P.O. BOX 388, HOLBROOK, N.Y. 11741

We are working on New Products and Software - CALL

We accept C.O.D. orders, money orders and ship within 24 Hours (most products). (Personal checks will have to clear before shipping.)



# LISTER PLUS

Reviewed by Peter Ellison

+ Lister Plus + boots up to a menu giving you a choice of three printer options. E-Epson, P-Prowriter, or O-Other. Choosing either 'E' or 'P' will put you in a menu with six different printing options. Picking 'O' will boot the Lister Plus Printer Drive Creator. This, then explains a little about printers, and how they receive data. It then runs a program that will ask different questions for different printed characters. It first asks the decimal equivalent to a certain character. Next it asks the number of columns your printer can reproduce. Finally, it asks for the code for Half dot columns. The printer driver is excellent, because you can create your own personal printer file, and use it any time by loading it from the Other file.

Going back to the main menu will give you a choice of six commands. They are 1) List a diskette file, 2) Type-A-Line file, 3) Print a Type-A-Line file, 4) Print a Graphics 8/7+ screen, 5) Print Character set tables, 6) Quit. The first option is used to print a program or text file. The program then asks if you wish 38 column format. This will print out your program or text file as it is read on the screen. You then have a choice of three widths: one, two, or three column. This can be handy when trying to debug a program. It then asks if you wish our listing Blocked Left, Centered, or Blocked right.

There is a special menu called General All Purpose Sub-Menu (Gaps) where you have six options. They are: 1) Standard Atari, this will give you the standard Atari font without anything fancy, 2) Whatever, is one where you can select your own current custom font for all printing, 3) New Custom, after choosing this option you can load in a new font, any of the many fonts that are given on the disk, or one that you've made up, 4) Line Feed (10), with this menu item, you can change the distance the printer will move after it prints the next line, 5) Ctrl/Alpha, you can replace all of the Graphics characters in the current custom font with Atari standard ALPHA characters (letters A to Z) in either upper case or lower case, 6) Main Menu, returns you back to the main menu.

Option number 2 in your main menu, Type-A-Line allows you to type a line and print it to your printer with whatever font you like. This makes it easy to put fancy headings, or whatever, on reports or letters. Option 3 is very similar to option 2, except you print a Type-A-Line file which you have saved before hand. Option 4, allows you to print a Graphics 8/7+ screen. Print either a micropainter screen or one of the many painting programs available. You can print it Inverse or Normal, or even on its side, using two sheets of paper (8X14). Option 5, Print Character Set

Tables, can be very useful when writing a program with a lot of character set. With this option you can print out any created character set, which makes it easy for quick reference. The final option is obvious, 6) Quit.

I recommend this utility to any one who owns a printer with a graphic chip, or one who is getting one. This utility, which is written in Basic, is easy to make a back-up copy, so don't abuse it. Starting with this issue we will be using this program to list our programs. This will make it a lot easier to type in our programs, because they'll look the same as on the screen, 38 columns. For \$19.95, how can you lose.

+ Lister Plus + is available on 48K disk from:

Non-Standard Magic  
P.O. BOX 45  
Girard, Ohio 44420

## NEW PRODUCT

### Enhancements to Basic

Despite its wide usage, Atari BASIC has a number of weaknesses. Unlike some other languages, BASIC lacks a SEARCH function and many other multi-line commands. Recently, FIRST BYTE has created a new utility to alleviate these problems. Called 'Enhancements to Basic', this program is stored to any disk, and is automatically booted when the computer is powered-up. As a result, the inconvenience of additional commands for LOADING are avoided. The additional functions provided by 'Enhancements to Basic' include commands to: Trace, renumber lines, list variables, automatically number lines, and delete blocks of data. In addition, most of the old DOS functions like Delete File, Examine File Directory, Rename File, Lock, Unlock, Erase File, and Format are incorporated into BASIC. Further, several new DOS commands, like Restore Deleted File, Examine Deleted File Directory, and Rename Deleted File are added. But what makes this utility particularly interesting is the relatively low price of \$14.95. Because ROM has received only a preliminary copy of 'ENHANCEMENTS TO BASIC', we have been unable to analyze this program in detail. However, once receiving a final copy, we plan to do a thorough review. (Check for it in the next issue.)

# Home & Business Programs

by Stephen Everett

The BASIC 'Home and Business' programs which accompany this text were contributed to ROM magazine by Stephen C. Everett of Victoria, B.C. Since completing his bachelor's degree in Business Administration, Stephen realized that many people could benefit from some of the knowledge he gained regarding the calculation of returns on moneys invested and the costs of financing. The program, which Stephen has provided, addresses thirteen types of financial calculations, outlined in more detail below. Although the calculations of the future value of a bond purchase on savings account deposit, or of the real cost of a mortgage, are tedious when performed by hand, this program acts as a 'smart business calculator' which allows you to determine the cost or value of money in a variety of common financial situations.

## Description of Programs

**Future Value of a Deposit:** Determines the future value of an investment, eg. bond or savings, given the initial investment, length of investment, annual interest rate, and the number of times interest is compounded annually.

**Future Value of an Annuity:** This program calculates the amount of each payment required to give an annuity with a fixed value in the future, eg. how much money to deposit each month to save \$25,000 in eight years. Input consists of the amount of money desired, annual interest rate, number of payments per year, and the number of years.

**Units of Output Depreciation:** Allows you to determine the depreciation per unit, based on initial cost, residual cost, and number of units.

**Straight-line Depreciation:** Calculates the current value, expense, and depreciation year, given the annual depreciation rate, the initial value, and the number of years to depreciate.

**Break-Even Analysis:** Calculates the number of units which must be produced to 'Break-Even', given the fixed manufacturing costs, costs of producing each unit, and the intended sales price per unit.

**Economic Order Quantity:** This program is used to determine the optional number of items to order at a given time, based on the cost of ordering, annual consumption of units, and the unit carrying cost.

**Weighted Average:** Calculates the weighted average of a series of values, given the values and the weights for each value.

**Payment Calculation:** Determines the total amount

paid for a loan, given the amount borrowed, annual interest rate, and the length of the loan in years.

**Mortgage Schedule:** Lists the amount of your mortgage payment which is applied to principal and interest charges, as well as the balance owing by month for the length of the mortgage. The program requests amount borrowed, annual interest rate, length of the loan, and the year for schedule.

**Remaining Balance:** Calculates the balance remaining on a mortgage, given the amount borrowed, annual interest, term of the loan, and the payment number.

**Mortgage with Second:** Gives the monthly payments for first and second mortgages, given the initial purchase price, cash on hand, amount of the first mortgage, and the interest rates and terms for both mortgages.

```
10 DIM A(15),B(15),C(15),D(15),E(15),F
(15),G(15),H(15),I(15),J(15),K(15),L(1
5),M(15),N(15)
20 DIM O(15),P(15),Q(15),R(15),S(15),T
(15),U(15),V(15),W(15),X(15),Y(15),Z(1
5),L$(10),O$(10)
30 DIM X$(10),Q$(10),W$(10),N$(10),M$(
10),G$(10),U$(10)
90 GOSUB 10000
100 ? "K"
110 REM -----
115 ? "      HOME AND BUSINESS PROGRAMS"
120 REM -----
121 ? "-----"
122 ?
123 ? "(1)  = EXIT FROM PROGRAM"
125 ? "(2)  = FUTURE VALUE OF A DEPOSIT"
130 ? "(3)  = FUTURE VALUE OF AN ANNUITY"
135 ? "(4)  = PRESENT VALUE OF AN AMOUNT"
140 ? "(5)  = UNITS OF OUTPUT DEPRECIATION"
145 ? "(6)  = DECLINING BALANCE DEPRECIATION"
150 ? "(7)  = STRAIGHT-LINE DEPRECIATION"
155 ? "(8)  = BREAK-EVEN ANALYSIS"
160 ? "(9)  = ECONOMIC ORDER QUANTITY"
165 ? "(10) = WEIGHTED AVERAGE"
170 ? "(11) = PAYMENT CALCULATION"
175 ? "(12) = MORTGAGE SCHEDULE"
180 ? "(13) = REMAINING BALANCE"
185 ? "(14) = MORTGAGE WITH SECOND"
190 ?
195 ? "DO YOU REQUIRE ANY FURTHER INFORMATIONON THE ABOVE PROGRAMS (Y/N)";:INPUT X$
200 IF X$="N" THEN 250
205 IF X$="Y" THEN 210
207 GO TO 250
210 ? "WHICH ONE":INPUT X$
220 IF X$="1" THEN 210
221 IF X$="2" THEN 15000
222 IF X$="3" THEN 16000
223 IF X$="4" THEN 17000
224 IF X$="5" THEN 18000
225 IF X$="6" THEN 19000
226 IF X$="7" THEN 20000
227 IF X$="8" THEN 21000
228 IF X$="9" THEN 22000
229 IF X$="10" THEN 23000
```

```

230 IF X$="11" THEN 24000
231 IF X$="12" THEN 25000
232 IF X$="13" THEN 26000
233 IF X$="14" THEN 27000
234 GOTO 210
250 ? "MAKE YOUR SELECTION";:INPUT X$
251 IF X$="1" THEN 5000
252 IF X$="2" THEN 300
253 IF X$="3" THEN 1000
254 IF X$="4" THEN 1200
255 IF X$="5" THEN 500
256 IF X$="6" THEN 700
257 IF X$="7" THEN 1500
258 IF X$="8" THEN 1700
261 IF X$="9" THEN 1800
262 IF X$="10" THEN 1900
263 IF X$="11" THEN 2000
264 IF X$="12" THEN 2100
265 IF X$="13" THEN 2600
266 IF X$="14" THEN 2900
270 GOTO 250
300 REM START
309 ? "K"
310 REM -----
320 ? " FUTURE VALUE OF A DEPOSIT"
325 REM -----
327 ?
328 ?
330 ? "INPUT DATA:"
335 ?
340 TRAP 340: ? "ORIGINAL DEPOSIT:";:IN
PUT M
350 ? "INTEREST RATE:";:INPUT S
355 ? "PERIODS PER YEAR:";:INPUT B
356 ? "NUMBER OF YEARS:";:INPUT C
365 ?
366 ?
368 LET S=S/100
369 LET S=S/B
370 LET S=INT((M*(1+S)^(B*C))*100)/100
380 ? "$";S;" IS THE FUTURE AMOUNT."
385 ?
386 ?
390 ? "ANY MORE CALCULATIONS";:INPUT G
$
400 IF G$="Y" THEN 309
410 GO TO 100
500 REM START
510 REM -----
519 ? "K"
520 ? " UNITS OF OUTPUT METHOD (DEP
R.)"
530 REM -----
535 ?
536 ?
540 ? "INPUT DATA:"
545 ?
550 TRAP 550: ? "COST OF UNIT:";:INPUT
C
555 ? "RESIDUAL VALUE:";:INPUT R
560 ? "ESTIMATED UNITS OF OUTPUT:";:IN
PUT E
570 LET T=INT(((C-R)/E)*100)/100
580 ?
585 ?
590 ? "$";T;" DEPRECIATION PER UNIT."
600 ?
605 ? "ANY MORE CALCULATIONS";:INPUT
M$
610 IF M$="Y" THEN 519
620 GO TO 100
700 REM START
701 ? "K"
710 REM -----
720 ? " DECLINING-BALANCE METHOD"
730 REM -----
739 ?
740 ? "ENTER DATA:"
741 ?
745 ? "NAME OF ITEM:";:INPUT N$
746 ? "NAME OF ITEM:";:INPUT N$
746 TRAP 746: ? "DEPRECIATION PERCENTAG
E:";:INPUT P
747 ? "BOOK VALUE:";:INPUT H
748 ? "YEARS TO DEPRECIATE:";:INPUT Y
749 ?
750 ? "ITEM:";:PRINT N$
753 ?
754 ?
755 ? " ", "DEP.", "ACC.", "BOOK"
756 ? "YEAR", "EXPENSE", "DEP.", "VALUE"
757 ?
760 A=0
855 LET P=P/100
859 LET X=1
862 LET D=INT((P*H)*100)/100

```

```

863 LET A=(D+A)
865 LET B=INT((H-(H*P))*100)/100
879 PRINT X,D,A,B
880 LET X=X+1
881 LET H=H-D
885 IF X>Y THEN 898
895 GO TO 862
898 ?
899 ?
900 ? "ANY MORE CALCULATIONS";:INPUT Q
$
910 IF Q$="Y" THEN 701
920 GO TO 100
1000 REM START
1010 ? "K"
1015 REM -----
1020 ? " FUTURE VALUE OF ANNUITY"
1025 REM -----
1026 ?
1027 ?
1030 ? "INPUT DATA:"
1031 ?
1035 TRAP 1035: ? "SIZE OF EACH PAYMENT
:";:INPUT T
1040 ? "INTEREST RATE:";:INPUT K
1041 ? "NUMBER OF YEARS:";:INPUT X
1045 ? "PAYMENTS PER YEAR:";:INPUT U
1046 ?
1047 ?
1048 LET K=K/100
1049 LET K=K/U
1050 LET S=INT((T*((1+K)^(X*U))-1)/K)
*100)/100
1060 ? "THE FUTURE VALUE IS $";S
1065 ?
1066 ?
1070 ? "ANY MORE CALCULATIONS";:INPUT
L$
1080 IF L$="Y" THEN 1010
1090 GO TO 100
1200 ? "K"
1210 REM -----
1200 ? "K"
1215 ? " PRESENT VALUE OF FUTURE AMOU
NT"
1220 REM -----
1230 ?
1231 ?
1232 ?
1240 ? "ENTER DATA:"
1241 ?
1245 TRAP 1245: ? "AMOUNT WANTED IN FUT
URE:";:INPUT Z
1250 ? "INTEREST RATE:";:INPUT V
1253 ? "PERIODS PER YEAR:";:INPUT F
1254 ? "NUMBER OF YEARS:";:INPUT X
1255 ?
1260 ?
1270 LET V=V/100
1271 LET V=V/F
1275 LET P=INT((Z*(1+V)^(F*X))*100)/1
00
1280 ? "PRESENT VALUE IS $";P
1285 ?
1286 ?
1290 ? "ANY MORE CALCULATIONS";:INPUT
O$
1300 IF O$="Y" THEN 1205
1310 GO TO 100
1500 REM START
1505 ? "K"
1510 REM -----
1515 ? " STRAIGHT-LINE DEPRECIAT
ION"
1520 REM -----
1522 ?
1523 ?
1524 ?
1530 ? "ENTER DATA:"
1531 ?
1535 TRAP 1535: ? "COST OF ITEM:";:INPU
T D
1540 ? "RESIDUAL VALUE:";:INPUT I
1550 ? "YEARS OF USEFUL LIFE:";:INPUT
J
1555 ?
1556 ?
1560 ? "YEAR", "DEP($)", "ACC. DEP.", "VAL
UE($)"
1563 ?
1570 LET T=INT(((D-I)/J)*100)/100
1573 LET Z=T
1575 LET A=1
1580 LET L=D-Z
1585 ? A,T,Z,L

```



```

1590 IF A=J THEN 1608
1591 LET Z=Z+T
1595 LET A=A+1
1596 GOTO 1580
1608 ?
1609 ?
1610 ? "ANY MORE CALCULATIONS";:INPUT
X$
1615 IF X$="Y" THEN 1505
1620 GO TO 100
1692 ?
1700 REM START
1705 ? "K"
1710 REM -----
1715 ? " BREAK-EVEN POINT"
1720 REM -----
1721 ?
1722 ?
1723 ?
1730 ? "ENTER DATA:"
1731 ?
1735 TRAP 1735: ? "FIXED COSTS";:INPUT
A
1738 ? "COST PER UNIT";:INPUT B
1741 ? "SALES PRICE";:INPUT C
1742 ?
1743 ?
1745 LET X=C-B
1748 LET T=INT((A/X)*100)/100
1750 ? "BREAK-EVEN POINT IS ";T;" UNIT
S."
1753 ?
1755 ?
1758 ? "ANYMORE CALCULATIONS";:INPUT Q
$
1760 IF Q$="Y" THEN 1705
1780 GOTO 100
1800 REM START
1801 ? "K"
1805 REM -----
1810 ? " ECONOMIC ORDER QUANTITY"
1815 REM -----
1817 ?
1818 ?
1819 ?
1820 ? "ENTER DATA:"
1821 ?
1825 TRAP 1825: ? "COST TO ORDER";:IMP
UT A
1830 ? "ANNUAL UNITS USED";:INPUT B
1835 ? "UNIT CARRYING COST";:INPUT C
1840 ?
1841 ?
1845 LET T=INT((5QR((2*B*A)/C))*100)/1
00
1850 ? "THE EOQ IS ";T;" UNITS."
1851 ?
1852 ?
1853 ?
1860 ? "ANY MORE CALCULATIONS";:INPUT
X$
1865 IF X$="Y" THEN 1801
1870 GO TO 100
1900 REM START
1905 ? "K"
1910 REM -----
1915 ? " WEIGHTED AVERAGE"
1920 REM -----
1922 ?
1923 ?
1924 ?
1925 ? "ENTER DATA, ALSO ENTER # OF VAL
UES:"
1926 ?
1930 TRAP 1930: ? "NUMBER OF VALUES (PA
IRS);:INPUT A
1935 ?
1936 LET S=0
1937 LET V=1
1939 ? "VALUE";:INPUT B
1940 ? "UNITS";:INPUT C
1941 ?
1942 ?
1951 LET F=B*C
1953 LET S=S+C
1955 LET T=T+F
1957 IF V=A THEN 1970
1960 LET V=V+1
1965 GOTO 1939
1970 LET Z=INT((T/S)*100)/100
1975 ? "WEIGHTED AVERAGE IS ";Z
1976 ?
1977 ?
1980 ? "ANY MORE CALCULATIONS";:INPUT
X$

```

```

1981 IF X$="Y" THEN 1905
1984 GO TO 100
2000 REM START
2005 ? "K"
2010 REM -----
2020 ? " PAYMENT CALCULATION"
2022 REM -----
2023 ?
2024 ?
2025 ?
2030 ? "ENTER DATA:"
2031 ?
2035 TRAP 2035: ? "AMOUNT BORROWED";:I
NPUT A
2040 ? "ANNUAL INTEREST RATE";:INPUT
B
2045 ? "TERM OF LOAN (YRS);:INPUT C
2046 ?
2047 ?
2050 LET B=B/100
2055 LET I=B/12
2060 LET T=INT((A/((1-((1+I)^(C*12)))
/I))*100)/100
2070 ? "PAYMENT IS $";T
2075 ?
2076 ?
2080 ? "ANYMORE CALCULATIONS";:INPUT
X$
2085 IF X$="Y" THEN 2005
2090 GO TO 100
2100 REM START
2105 ? "K"
2110 REM -----
2112 ? " MORTGAGE SCHEDULE"
2113 REM -----
2121 ?
2122 ?
2123 ?
2130 ? "ENTER DATA:"
2135 ?
2140 TRAP 2140: ? "AMOUNT BORROWED";:I
NPUT A
2145 ? "ANNUAL INTEREST RATE";:INPUT
B
2150 ? "TERM OF LOAN (YRS.);:INPUT C
2155 ? "YEAR FOR SCHEDULE";:INPUT D
2160 ?
2165 ?
2166 LET L=1
2167 LET D=(D-1)*12
2168 LET B=B/100
2169 LET B=B/12
2170 LET MP=INT((A/((1-((1+B)^(C*12))
)/B))*100)/100
2175 LET T=INT((MP*((1-((1+B)^(C*12)
-D))/B))*100)/100
2185 ? "MONTH", "PRINC.", "INTEREST", "BA
LANCE"
2189 LET Z=INT((T*B)*100)/100
2190 LET X=MP-Z
2198 ?
2199 LET T=T-X
2200 ? D, X, Z, T
2218 LET D=D+1
2222 IF L=13 THEN 2300
2223 LET L=L+1
2230 GO TO 2189
2300 ? "ANY MORE CALCULATIONS";:INPUT
X$
2310 IF X$="Y" THEN 2105
2320 GO TO 100
2600 REM START
2605 ? "K"
2610 REM -----
2615 ? " REMAINING BALANCE"
2620 REM -----
2622 ?
2623 ?
2624 ?
2630 ? "ENTER DATA:"
2632 ?
2635 TRAP 2635: ? "AMOUNT BORROWED";:I
NPUT A
2640 ? "ANNUAL INTEREST RATE";:INPUT
B
2645 ? "TERM OF LOAN (YRS);:INPUT C
2650 ? "PAYMENT NUMBER";:INPUT D
2658 ?
2660 ?
2666 ?
2670 LET I=B/100
2673 LET I=I/12
2675 LET T=INT((A/((1-((1+I)^(C*12))
/I))*100)/100
2680 ? "MONTHLY PAYMENT $";T

```

```

2685 LET S=INT((T*((1-((1+I)^(-(12*C)-
D))/I))*100)/100
2690 ? "BALANCE AFTER ";D;" PAYMENTS I
S $";S
2693 ?
2694 ? "ANYMORE CALCULATIONS";:INPUT X
S
2695 IF X$="Y" THEN GO TO 2605
2699 GO TO 100
2900 REM START
2905 ? "K"
2910 REM -----
2915 ? " MORTGAGE PAYMENT WITH SECOND"
2920 REM -----
2921 ?
2922 ?
2923 ?
2925 ? "ENTER DATA:"
2926 ?
2930 TRAP 2930: ? "PURCHASE PRICE:";:IN
PUT A
2935 ? "CASH AVAILABLE:";:INPUT B
2940 ? "FIRST MORTGAGE:";:INPUT C
2945 ? "FIRST RATE (%):";:INPUT D
2950 ? "FIRST TERM (YRS):";:INPUT E
2960 ? "SECOND RATE (%):";:INPUT G
2962 ? "SECOND TERM:";:INPUT U
2963 ?
2964 ?
2970 LET D=D/100
2971 LET I=D/12
2975 LET G=G/100
2976 LET K=G/12
2980 LET X=A-B
2985 LET P=INT((C/((1-((1+I)^-(12*E))
/I))*100)/100
2990 ? "FIRST PAYMENT $";P
2995 LET L=X-C
3000 LET M=INT((L/((1-((1+K)^-(U*12))
/K))*100)/100
3005 ? "SECOND PAYMENT $";M
3007 ?
3010 LET T=P+M
3015 ? "TOTAL PAYMENTS $";T
3016 ?
3017 ?
3018 ?
3020 ? "ANYMORE CALCULATIONS";:INPUT N
S
3025 IF N$="Y" THEN 2905
3333 GO TO 100
5000 GRAPHICS 2+16
5010 PLOT 0,0:DRAWTO 19,0:DRAWTO 19,11
:DRAWTO 0,11:DRAWTO 0,0
5020 POSITION 1,1: ? #6;"PRACTICAL PROG
RAMS"
5030 POSITION 5,3: ? #6;"DESIGNED BY:"
5040 POSITION 1,6: ? #6;"STEPHEN C. EVE
RETT"
5050 POSITION 3,10: ? #6;"COPYRIGHT 198
4"
5060 GOTO 5060
10000 GRAPHICS 2+16
10005 COLOR 4
10010 PLOT 0,0:DRAWTO 19,0:DRAWTO 19,1
1:DRAWTO 0,11:DRAWTO 0,0
10020 POSITION 1,1: ? #6;" HOME AND BUS
INESS"
10030 POSITION 6,3: ? #6;"PROGRAMS."
10040 POSITION 5,5: ? #6;"DESIGNED BY"
10050 POSITION 1,7: ? #6;"STEPHEN C. EV
ERETT"
10060 POSITION 3,10: ? #6;"COPYRIGHT 19
84."
10070 IF PEEK(53279)=6 THEN RETURN
10080 GOTO 10070
15000 ? "K"
15010 ? "***** FUTURE VALUE OF A DEPOS
IT *****"
15020 ?
15021 ?
15030 ? " THIS PROGRAM ALLOWS THE
USER TO FIND THE FUTURE AMOUNT OF A D
EPOSIT."
15040 ? "ALL THAT IS NEEDED IS THE AMO
UNT OF THE DEPOSIT,THE ANNUAL INTERE
ST RATE,"
15050 ? "PERIODS PER YEAR, WHICH IS TH
E NUMBER OF TIMES INTEREST IS CALCULAT
ED EACH"
15055 ? "YEAR, ALSO THE NUMBER OF YEAR
S THE INTEREST WILL BE CALCULATED F
OR."
15060 ?
15061 ?

```

```

15062 ? "*****
*****"
15064 ?
15065 ? "PRESS THE START KEY WHEN READ
Y."
15070 IF PEEK(53279)=6 THEN 300
15071 GOTO 15070
16000 ? "K"
16010 ? "***** FUTURE VALUE OF AN ANNU
ITY *****"
16012 ?
16013 ?
16020 ? " THIS PROGRAM PERMITS THE
USER TO"
16025 ? "RESOLVE THE FUTURE VALUE OF A
SERIES"
16030 ? "OF DEPOSITS. ALL THAT IS REQ
UIRED IS THE SIZE OF EACH PAYMENT,THE
ANNUAL"
16040 ? "INTEREST RATE, THE NUMBER OF
YEARS THEDEPOSITS WILL CONTINUE FOR, A
ND THE NUMBER OF PAYMENTS PER YEAR."
16050 ?
16052 ?
16055 ? "*****
*****"
16060 ? "PRESS THE START KEY WHEN READ
Y."
16070 IF PEEK(53279)=6 THEN 1000
16080 GOTO 16070
17000 ? "K"
17010 ? "***** PRESENT VALUE OF AN AMO
UNT *****"
17020 ?
17021 ?
17025 ? " THIS PROGRAM ALLOWS THE
USER TO DETERMINE HOW MUCH MONEY IS N
EEDED"
17030 ? "PRESENTLY TO HAVE A CERTAIN S
UM IN THEFUTURE. ALL THAT IS NEEDED IS
THE"
17040 ? "AMOUNT NEEDED IN THE FUTURE,
THE ANNUAL INTEREST RATE, PERIODS
/YEAR,"
17050 ? "WHICH IS THE NUMBER OF TIMES
THE INTEREST IS CALCULATED EACH Y
EAR, AND"
17060 ? "THE NUMBER OF YEARS UNTIL THE
SUM OF MONEY IS REQUIRED."
17070 ?
17075 ?
17080 ? "*****
*****"
17085 ?
17090 ? "PRESS THE START KEY WHEN READ
Y."
17095 IF PEEK(53279)=6 THEN 1200
17099 GOTO 17095
18000 ? "K"
18010 ? "***** UNITS OF OUTPUT DEPRECI
ATION *****"
18013 ?
18015 ?
18020 ? " FOR CERTAIN TYPES OF ASSE
TS IT IS MUCH EASIER TO MEASURE DEPREC
IATION"
18030 ? "IN UNITS RATHER THAN TIME. T
HEREFORE,THIS METHOD OF DEPRECIATION I
S USED,"
18040 ? "ALL THAT IS REQUIRED IS THE C
OST OF THE ITEM, THE RESIDUAL OR SAL
VAGE"
18050 ? "VALUE, AND THE ESTIMATED UNIT
S OF OUTPUT."
18055 ?
18057 ?
18060 ? "*****
*****"
18065 ?
18070 ? "PRESS THE START KEY WHEN READ
Y."
18075 IF PEEK(53279)=6 THEN 500
18080 GOTO 18075
19000 ? "K"
19010 ? "**** DECLINING BALANCE DEPRECI
ATION ****"
19014 ?
19016 ?
19020 ? " THIS METHOD OF RAPID WRIT
E-OFF OF DEPRECIABLE ASSETS CONSISTS I
N THE"
19030 ? "DOUBLING OF THE NORMAL RATE.
WHAT IS REQUIRED IS THE NAME, OPTIONA
L,"
19040 ? "PERCENTAGE BY WHICH YOU WISH

```

TO DEPRECIATE BY, THE BOOK VALUE  
 , NET AMOUNT AT WHICH THE ASSET IS"  
 19050 ? "SHOWN IN THE ACCOUNTING RECOR  
 DS, AND THE NUMBER OF YEARS YOU WISH  
 TO DEPRECIATE IT."  
 19060 ?  
 19065 ?  
 19070 ? "\*\*\*\*\*"  
 19080 ?  
 19085 ? "PRESS THE **START** KEY WHEN READ  
 Y."  
 19090 IF PEEK(53279)=6 THEN 700  
 19095 GOTO 19090  
 20000 ? "K"  
 20010 ? "\*\*\*\*\* STRAIGHT-LINE DEPRECIATI  
 ON \*\*\*\*\*"  
 20015 ?  
 20016 ?  
 20020 ? " THIS IS THE SIMPLEST AND  
 MOST WIDELY USED METHOD OF DEPRECI  
 ATION."  
 20030 ? "ALL THAT IS NEEDED IS THE COS  
 T OF THE ITEM, THE RESIDUAL OR SALVAGE  
 VALUE,"  
 20040 ? "AND THE NUMBER OF YEARS YOU W  
 ANT TO DEPRECIATE IT BY."  
 20050 ?  
 20055 ?  
 20058 ? "\*\*\*\*\*"  
 20060 ?  
 20070 ? "PRESS THE **START** KEY WHEN READ  
 Y."  
 20080 IF PEEK(53279)=6 THEN 1500  
 20085 GOTO 20080  
 21000 ? "K"  
 21001 ? "\*\*\*\*\* BREAK-EVEN ANALYSIS  
 \*\*\*\*\*"  
 21010 ?  
 21011 ?  
 21020 ? " THIS IS THE STUDY OF PROFI  
 T-VOLUME RELATIONSHIPS. IT IS THE POI  
 NT AT"  
 21030 ? "WHICH LOSSES TURN INTO PROFIT  
 S. IT IS A VERY USEFUL TOOL FOR A MANA  
 GER TO"  
 21040 ? "CONTROL. WHAT IS REQUIRED IS  
 THE FIXED COSTS, ie. RENT etc., T  
 HE COST"  
 21050 ? "OF EACH UNIT TO THE COMPANY,  
 AND THE SALES PRICE."  
 21060 ?  
 21065 ?  
 21070 ? "\*\*\*\*\*"  
 21075 ?  
 21080 ? "PRESS THE **START** KEY WHEN READ  
 Y."  
 21085 IF PEEK(53279)=6 THEN 1700  
 21090 GOTO 21085  
 22000 ? "K"  
 22010 ? "\*\*\*\*\* ECONOMIC ORDER QUANTIT  
 Y \*\*\*\*\*"  
 22011 ?  
 22012 ?  
 22022 ? " THIS PROGRAM ALLOWS THE  
 USER TO ESTABLISH THE MOST EFFICIENT  
 QUANTITY"  
 22030 ? "TO ORDER. ALL THAT IS NEEDED  
 IS THE COST TO ORDER, ie. SERVICE CHA  
 RGE,"  
 22040 ? "THE ANNUAL UNITS THAT ARE USE  
 D BY THE COMPANY, AND THE COST/UNIT TO  
 STORE."  
 22050 ?  
 22055 ?  
 22060 ? "\*\*\*\*\*"  
 22065 ?  
 22070 ? "PRESS THE **START** KEY WHEN READ  
 Y."  
 22075 IF PEEK(53279)=6 THEN 1800  
 22080 GOTO 22075  
 23000 ? "K"  
 23010 ? "\*\*\*\*\* WEIGHTED AVERAGE \*\*  
 \*\*\*\*\*"  
 23020 ?  
 23022 ?  
 23030 ? " THIS IS A VERY BASIC PROG  
 RAM WHICH GIVES THE WEIGHTED AVERAGE.  
 IT ASKS"  
 23040 ? "FOR NUMBER OF VALUES, ie. 10 @  
 \$50.00, AND 5 @ \$10.00, THOSE ARE TWO  
 PAIRS."  
 23045 ? "IT THEN ASKS FOR THE VALUE WH

ICH IS THE \$50.00, AND THE \$10.00, TH  
 EN IT"  
 23050 ? "ASKS FOR THE UNITS WHICH IS T  
 HE 10 AND THE 5."  
 23055 ?  
 23056 ?  
 23060 ? "\*\*\*\*\*"  
 23065 ?  
 23070 ? "PRESS THE **START** KEY WHEN READ  
 Y."  
 23075 IF PEEK(53279)=6 THEN 1900  
 23080 GOTO 23075  
 24000 ? "K"  
 24010 ? "\*\*\*\*\* PAYMENT CALCULATION \*  
 \*\*\*\*\*"  
 24020 ?  
 24022 ?  
 24030 ? " THIS PROGRAM FIGURES OUT  
 YOUR MONTHLY PAYMENT ON A LOAN. A  
 LL THAT"  
 24040 ? "IS NEEDED IS AMOUNT BORROWED,  
 THE ANNUAL INTEREST RATE AND THE  
 TERM IN"  
 24050 ? "YEARS."  
 24060 ?  
 24065 ?  
 24070 ? "\*\*\*\*\*"  
 24075 ?  
 24080 ? "PRESS THE **START** KEY WHEN REA  
 DY."  
 24085 IF PEEK(53279)=6 THEN 2000  
 24090 GOTO 24085  
 25000 ? "K"  
 25010 ? "\*\*\*\*\* MORTGAGE SCHEDULE \*\*  
 \*\*\*\*\*"  
 25012 ?  
 25015 ?  
 25020 ? " THIS PROGRAM GIVES YOU A  
 N AMORTIZATION SCHEDULE FOR ANY  
 YEAR"  
 25030 ? "OF THE TERM OF THE MORTGAGE.  
 ALL THAT IS NEEDED IS THE AMOUNT  
 OF THE"  
 25040 ? "MORTGAGE, THE ANNUAL INTEREST  
 RATE, THE TERM OF THE LOAN, AND THE  
 YEAR"  
 25050 ? "IN WHICH YOU WANT THE SCHEDUL  
 E FOR."  
 25055 ?  
 25056 ?  
 25060 ? "\*\*\*\*\*"  
 25065 ?  
 25070 ? "PRESS THE **START** KEY WHEN READ  
 Y."  
 25075 IF PEEK(53279)=6 THEN 2100  
 25080 GOTO 25075  
 26000 ? "K"  
 26010 ? "\*\*\*\*\* REMAINING BALANCE \*\*  
 \*\*\*\*\*"  
 26020 ?  
 26022 ?  
 26030 ? " THIS PROGRAM FIGURES OUT  
 HOW MUCH MONEY IS LEFT TO PAY ON YOUR  
 LOAN OR"  
 26040 ? "MORTGAGE. ALL THAT IS NEEDED  
 IS THE AMOUNT BORROWED, ANNUAL INTERE  
 ST RATE,"  
 26050 ? "THE TERM OF THE LOAN IN YEARS  
 , AND THE PAYMENT NUMBER, WHICH IS  
 ALL THE"  
 26060 ? "PAYMENTS THAT HAVE BEEN PAID  
 ADDED"  
 26061 ? "TOGETHER."  
 26064 ?  
 26066 ?  
 26070 ? "\*\*\*\*\*"  
 26075 ?  
 26080 ? "PRESS THE **START** KEY WHEN READ  
 Y."  
 26085 IF PEEK(53279)=6 THEN 2600

Continued on Page 62



# Examining Atari DOS

by Bob Cockroft

The following is an in depth explanation of how the disk drive manipulates data. In order to accurately display a number of concepts, I wrote 2 utilities: the File Finder and the Disk Modifier. Both of these programs are listed at the end of the article. Because I will be constantly referring to them, it would be helpful for you to type them in now.

The disk is divided into three data organizing structures. The first and smallest structure is called the byte. This unit is used as a storage slot for one 8 bit number. In order to hold relatively large amounts of data, the disk contains thousands of these bytes. The second data organizing structure is the sector. It consists of 128 continuous bytes, positioned as a group at one specific location on the disk. Sectors are grouped into 40 concentric rings around the center of the disk. Being the last data structure, these rings are called tracks. Each of the 40 tracks contain 18 sectors for a total of 720 sectors per disk. Sectors 1 through 18 use the outermost track, while sectors 712 through 720 use the innermost.

When a disk is being formatted, the head will move sequentially through the tracks, reorganizing the sectors. The DOS will then reserve a number of sectors for particular tasks. Two important reserve sections are the Disk Boot Records, and the Table of Contents.

## Disk Boot Record

The Disk Boot Records are stored in sectors 1 through 3. When the Atari computer is powered-up, the operating system reads sector 1, to receive its initial instruction. It is here the computer encounters the Disk Boot Record. Byte 0 of the Disk Boot Record contains no instruction(see Table 1 below). This byte is a boot flag that always equals zero. Byte 1 tells the computer how many consecutive sectors it is to read. Under most circumstances this byte has a value of '3' to represent the 3 sectors of the Boot Record. Bytes 2 and 3 specify the memory address where the Boot Record is to be stored. This value is set to \$700 hex (1792 dec). Because the least significant byte always uses the first address, bytes 2 and 3 will have values of '0' and '7' respectively.

example:  $1792 = 0 + 7 * 16^2$

Therefore, because of the instructions given in the first 5 bytes of the Boot Record, when the computer is

powered-up, the operating system will read the first 3 sectors and store them into memory address \$700 hex (1792 dec). This is called the first stage of the boot. A list of the next few bytes of the Boot Records are listed below.

**Table 1**  
**Disk Boot Record**

Byte	Hex	Value	Description
0	700	0	Boot flag always zero
1	701	3	No. of consecutive sectors to read
2	702	0	Boot Record memory address
3	703	7	
4	704	64	Initialization address
5	705	21	
6	706	76	JMP XBCONT:Boot continuation
7	707	20	
8	708	7	
9	709	3	Max number of OPEN files
10	70A	3	Max number of Disk Drives
11	70B	0	Unused
12	70C	124	Buffer allocation address
13	70D	26	
14	70E	1	DOS flag: a non-zero indicates the disk contains a DOS file. One indicates a 128 byte sector disk, two indicates a 256 byte disk.
15	70F	4	Points to the first sector of the DOS.SYS file
16	710	0	
17	711	125	Sector link byte - points to the next sector to be read. It contains a zero when the end of the file has been reached
18	712	203	The start address of the DOS.SYS
19	713	7	

The second stage of the boot loads the File Management System of the DOS into memory. Most software does not use the second stage of the boot. The first byte of sector 1 would have a sector count that represented the entire program. In this way, software can be loaded before the computer is able to perform any other function.

## Directory

The directory is the area on the disk the computer designates for organizing the files. Stored on 9 sectors (360-368), this structure represents the single most important part of the disk. The directory can be divided into two sections: Table of Contents and File Directory.

The Table of Contents is used by the DOS to keep track of which sectors contain file data and how much free space remains. The file structure of the entire disk is squeezed on to sector 360. To examine the Table of Contents, display sector 360 in hexadecimal form with the Files Finder. (see below)

**Table 2**

bit:	0	1	2	3	4	5	6	7
HEX	00	02	C3	02	C1	01	00	00
08	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00
18	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	0F	FF
28	FF	FF	FF	FF	FF	FF	FF	FF
30	FF	00	00	00	00	3F	FF	00
38	7F	FF	FF	FF	FF	FF	FF	FF
40	FF	FF	FF	FF	FF	FF	FF	FF
48	FF	FF	FF	FF	FF	FF	FF	FF
50	FF	FF	FF	FF	FF	FF	FF	FF
58	FF	FF	FF	FF	FF	FF	FF	FF
60	FF	FF	FF	FF	00	00	00	00
68	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00
78	00	00	00	00	00	00	00	00

The bytes 1 and 2 specify the maximum number of sectors that are able to be used for data storage. These bytes use a hi/lo form and will normally contain a value of \$2C3 hex (707 dec.). Bytes 3 and 4 indicate the number of free sectors remaining on the disk. On an empty disk these bytes will have a value of \$2C3 hex (707 dec.), on a full disk they will have a value of \$00. Each bit, from byte \$0A to \$63 corresponds to a particular sector. For example, byte \$0A represents sectors 1 and 2, while byte \$63 represents sectors 711 through 719. Sector 720 is not used by the DOS so that a buffer zone exists between the last usable sector and the end of the disk. Initially these sector bits are set to '1'. But as a file is stored on the disk, the bit corresponding to the sector where file is written will be given a value of '0'. When a file is deleted, the bits are set back to '1'

By referring to the Table of Contents, the DOS is always able to determine which sectors are currently being used. This is of critical importance, particularly

F = Free

U = Used

Sector	A	A+1	A+2	A+3	A+4	A+5	A+6	A+7
Fr//Usd	F	F	U	F	U	F	F	F
bit:	0	1	2	3	4	5	6	7
value:	1	1	0	1	0	1	1	1

$$235 = 1 + 2 + 0 + 8 + 0 + 32 + 64 + 128$$

Therefore: this byte will have a value of 235

when the computer is storing files on the disk. You may have noticed that even on a newly formatted disk, sectors 1,2 and 3 (bits 6,5,4 of byte 0A\$=0F\$) always are occupied. The reason for this is that these first 3 sectors contain the Disk Boot Record. In addition, a group of 9 bytes, beginning at byte \$37, are also permanently used. These bytes correspond to the sectors that are occupied by the directory (sectors 360-368).

The second component of the Directory is the File Directory. Stored in sectors 361 through 368, this data structure contains a list of all the files on the disk along with their status, starting sector and size. Examine sector 361 with the File Finder while in character mode. If all is working correctly you should see the names of some, or perhaps all the files on your disk. Notice that file data is contained in an entry of 16 bytes. The first letter of the filename begins on the 6th byte of the entry, while the extension starts on the 12th. If any of the character bytes of the filespec are unused, they are left blank. The File Directory can use 8 sectors of 128 bytes a piece, for a total of 1024 bytes(8\*128). Because each file uses 16 bytes, the File Directory can hold a maximum of 64 filenames(1024/16). Re-examine sector 361 with the File Finder, only this time use the hexadecimal mode. The first byte of each 16 byte entry is the status flag for the file. An unlocked file will set the status flag to \$42(66 dec.), whereas a locked file has a status of \$62(98 dec). If a file were deleted it would have a status of \$80(128 dec). The second and third bytes of an entry display the file size in sectors. The fourth and fifth bytes point to the first sector of the file.

### 16 byte configuration

byte no.	1	2	3	4	5	6	7	8	...
Value	:42	:01	:00	:04	:00	:54	:45	:53	:...cont

Status: = \$42 = Unlock file;byte 1

Size: = \$01 = 1 sector ;byte 2 & 3

First Sector: = \$04 = sector 4 ;byte 4 & 5

Bytes 6 through 16 contain the filename

## Linking and File Organization

If the Table of Contents of a disk which has had a large number of file additions and deletions were seen, the programs would appear to be quite fragmented. One file could start on sector 25 but then jump to sectors 30, 50, and 275. Because of 'linking', files do not need to use sectors in numerical order. As a result, even sectors isolated by other files can be used to store data. When a new file is SAVED, the DOS creates a corresponding 16 byte entry in the Directory. This entry includes the filename and status(\$43). The DOS will then search the Table of Contents for a free sector. Once a free sector has been found, its number is stored in the fourth and fifth bytes of the 16 byte entry as the starting sector of the file.(see below)

### 16 Byte Entry

Values in Hexadecimal

Byte no.	0	1	2	3	4	5	6	7	...
Value	43	00	00	09	00	14	49	15	...cont
	:	:	:	:	:	:	:	:	
	status	:	:	first	:	start of	:	:	
	:	:	:	size	:	sector	:	filename	
	:	:	:	:	:	:	:	:	
	:	:	:	:	:	:	:	:	

The first sector of this file  
is sector 9

Data is then stored on the first 124 bytes in the start sector. The 3 remaining bytes(125 - 127) are used by the DOS to direct or 'link' itself to the next sector the file will use. The low 6 bits of byte 125 indicate the file identification number. The DOS uses this to determine which file a sector belongs to. The value of the identification number represents the position the file has in the directory(0-63). Should the value not correspond to the order in which it appears in the File Directory, the DOS will produce an error. The two high bits of byte 125 and all of byte 126 are reserved so that they can point to the next sector of the file. The DOS will then scan the Table of Contents for the next free sector. Once a free sector has been located, its number is stored in bytes 125 and 126 of the previous sector used for data storage. In this way, each sector of a file is linked to the next. What makes the sector link is usually the way in which it stores data in 2

bytes. Most 2 byte configurations store data in lo/hi form, in other words, low byte first, and high byte second. For example, \$600 hex would be stored as follows:

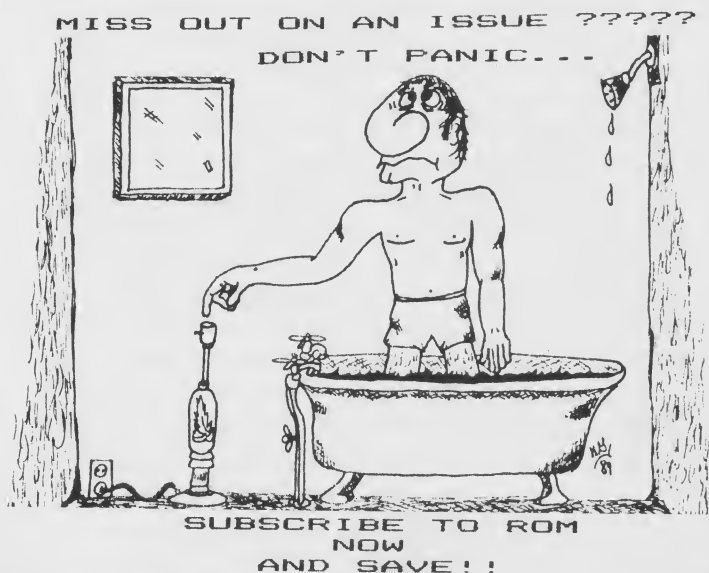
Memory location	Value
1536	0 low byte
1537	6 high byte

The sector link does not use 2 bytes in this manner. Instead, the value is stored in a modified high byte/low byte form. Byte 126 is used as the low byte value, whereas bits 6 and 7 of byte 125 are used as the high values. When the last sector of a file has been reached, the sector pointers will indicate this by having a value of '0'. The last part of the sector link, byte 127, contains the number of bytes used in that sector. In most situations this byte will have a value of \$7D(125 dec). The only exception would be the last sector of a file, where it is probably only partly filled. Only when the user closes a file will this partial sector be filled. A diagram of the sector link operations is given below.

Decimal version

Byte No.	Value	;Comment
125	32 (\$20)	;32th Directory file
126	25 (\$18)	;Sector 25 is the next ;sector in the file
127	125 (\$7D)	;125 of bytes used ;from the current sector

Use the File Finder to examine sectors that have a file stored on them. The link bytes(last 3 bytes) of any of the sectors you examine should have a display similar to the hexadecimal output given below. In addition, compare the values of the decimal version(above) with the hexadecimal version(below).





## Hexadecimal version

bit:	0	1	2	3	4	5	6	7
HEX								
00	C3	04	09	A0	C3	45	40	01
08	00	00	AA	D0	01	10	BF	25
10	35	99	63	A5	C2	00	05	75
18	21	56	5D	FF	FF	EF	A0	A9
20	10	00	01	12	15	A3	A2	05
28	07	05	A3	C2	04	47	59	84
30	38	FF	EA	47	36	84	30	38
38	48	57	95	04	57	48	04	48
40	FF	A4	47	47	39	E4	D3	FF
50	47	12	20	03	14	69	48	04
58	E6	48	FF	EA	A9	47	A2	47
60	47	07	35	13	03	93	03	37
68	E4	FF	AA	33	37	C3	B4	47
70	B4	47	A3	47	99	87	44	34
78	47	A9	10	48	72	20	18	7D

these.  
last 3 bytes  
are the link bytes

After the disk file has been closed, The DOS will write an up-dated version of the Table of Contents, which includes all the newly occupied sectors. It then up-dates the 16 byte File Directory entry by changing its status to \$42(normal status), and by storing the file size(in sector) into bytes 1 and 2.(see below)

### 16 byte entry (values in HEX)

Byte no.	0	1	2	3	4	5	6	7	...
Value	:42	:05	:00	:09	:00	:14	:A9	:15	...cont
	:	:	:	:	:	:	:	:	
	status:		first:		start of				
	:size		:sector		: filename				

When a LOAD command is made, the DOS will first search the File Directory for the 16 byte entry of the specified file. Once it is found, the DOS will use bytes 3 and 4 to determine its first sector. Once the data has been read off this sector, the link bytes will point the DOS to the next sector of the file. This process of 'linking' is repeated until all the sectors of the file have been read.

## Types of Files

The 3 main methods in which data is stored on the disk are untokenized, tokenized, and binary systems. An untokenized file is created whenever a file is stored on the disk by using the LIST command. This type of system does not use any method of reducing its size. Because a file of this type is stored in its

original format, it will appear the same on the disk as it did while in the computer. Create an untokenized file and then use the File Finder's character mode to examine the sectors of this file. The file will appear recognizable. You should be able to read it off the screen as you could under normal circumstances.

The tokenized file is created whenever a file is stored on the disk using the SAVE command. This type of arrangement uses a reducing system to modify the file so that it can be loaded into the computer more rapidly. As a result, the file appears, at best, barely recognizable when examined by the File Finder. In fact, only the PRINT and REM commands appear normally.

The last main type data organization system is the binary file. This type of storage method is generally used in conjunction with the DOS's BINARY LOAD/SAVE options in order to transfer machine code between the disk and memory. Create a binary file, and store it on the disk. Because a binary file does not need to be a workable machine language routine, use the BINARY SAVE option to store any portion of RAM. Once this is done, use the hexadecimal mode of the File Finder to examine the first sector of this file. Binary files start with two bytes of \$FF. The next 2 bytes store in lo/hi form, the start address of the file. The 2 bytes that follow these store in lo/hi form the end address. When a binary load is completed, control is passed to the DOS menu. However, after the loading sequence, the DOS can be forced to pass control to any address by storing that address in locations \$2E0 and \$2E1. For example, if your binary file started a \$600, store a \$00 into address \$2E0 and a \$06 into address \$2E1. As a result the BINARY LOAD option will both load and run the file.

## Sector Modifying Utility

If you have accidentally erased a file that you needed, this utility may solve your problem. By making a simple modification to the File Directory, it is possible to retrieve a deleted file. When a file is erased, the DOS performs two functions. The first is to go to the Table of Contents, and set the bits that correspond to sectors used by the file, to '1'. All this does is tell the DOS that it is now able to use these sectors. The second function is to set the file's status flag to \$80(delete mode). The significant thing, is that despite telling the DOS to delete, the file remains on the disk. Assuming programs that were stored after the original file was erased did not write over the deleted sectors, retrieving a file is simply a matter of finding it, and changing the status flag. This type of byte modification is exactly what the following program does. This would be a good time to type-in, or if you have the disk version, LOAD the Sector

Modifier. Soon after RUNing this utility, it will ask you to identify the sector that is to be changed. Once the program has this input, the sector you identify will be loaded into the memory(1664-1792 dec) and a new screen containing 4 options will appear. The first two of these options control whether the stored sector will be displayed in hexadecimal or character form. The character display is useful when examining File Directory sectors because the filenames can be read off the screen. However, for the majority of applications, the hexadecimal display is the most convenient. The third option enables the user to choose another sector to examine. The fourth option is where any disk modification occurs. It does much the same thing for the disk as the POKE command does for memory. The value of any individual byte on the disk can be altered. All the utility will ask for is the byte number(relative to the start of the sector) and the new value. The modification to the sector is not made initially. The new value is POKEd into the memory locations that contain the copy of the sector (1664-1792 dec). This memory held sector is then stored on the disk over-top of the old sector.

### Disk Routines

In order to make the Sector Modifier I had to write a machine language subroutine to LOAD data from the disk, and another to SAVE data to the disk. Since that time, I have used these routines in a number of other machine language programs that require data transfer. Many programs, especially adventure games, can be enlarged by using the disk to storage surplus data. For example, the computer could have access to a number of scrolling map routines by storing the data for each display on disk. When a particular map is required, its data can be LOADED to computer. As a result, programs that do not need quick access to all of its subroutines are not confined to the computer's RAM limitations. By using many disks, programs can be as large as desired. For this reason, machine language subroutines that LOAD and SAVE are quite useful if applied effectively. At the same time, routines of this type are simple to make.

Many of the addresses that are used to control the disk drive are located from \$300 to \$30B HEX (768 to 779 dec). In order to provide a quick reference to the relevant addresses, I have listed them below.

#### Location

HEX	DEC	LABEL	;FUNCTION
301	769	DUNIT	;Device number
302	770	DCOMND	;No. of operation

304	772	DBUFLO	;Address of the source
305	773	DBUFHI	;destination of data
30A	778	DAUX1	;Disk sector number
30B	779	DAUX2	;for read and write operations

There are 5 steps in making a LOAD or SAVE subroutine. First, before any disk operation can occur, the drive number must be specified. This is done by storing the disk drive number at address \$301 DUNIT(769 dec).

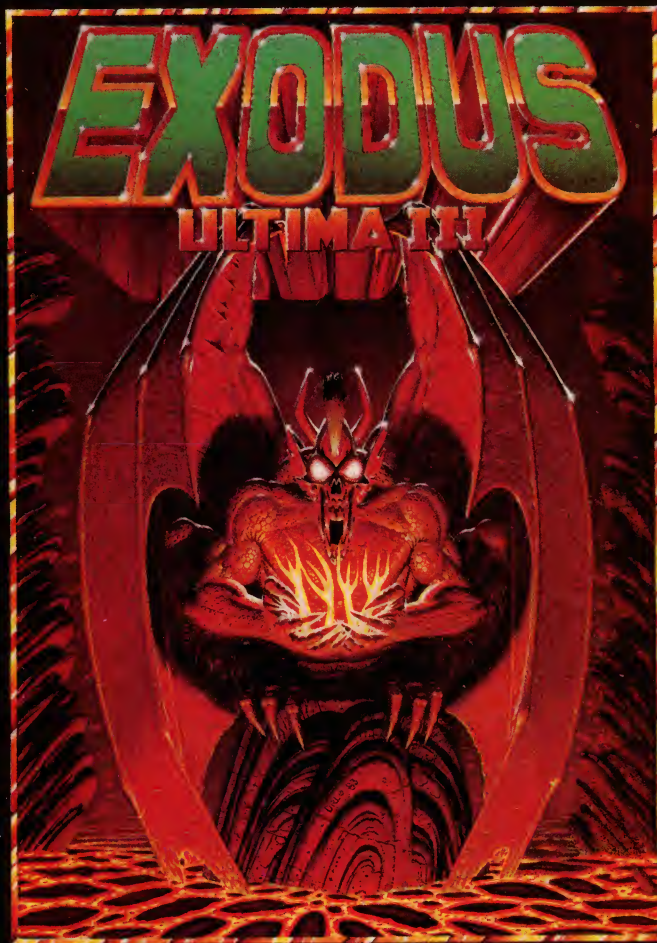
The second step is to indicate what operation the disk drive is to perform. The disk operation is controlled by address \$302 DCOMND (770 dec). This location reacts to values that correspond to specific commands.(see below)

Command	dec	hex
Read (LOAD)	82	\$52
Write (SAVE)	87	\$57
Download	32	\$20
Format	33	\$21
Put	80	\$50
Read spin	81	\$51
Status	83	\$53
Read address	84	\$54
Verify sector	86	\$56

For example, if you wish to LOAD data from the disk to the computer, you would store '82' into \$302(DCOMND). Conversely, if you wished to SAVE data, a '87' would be stored here. The same system is used for the rest of the commands.

The third step is to decide what RAM address the data is to be transferred, either to or from. When the DCOMND (\$302 hex) is given a Read(LOAD) command, the DOS refers to locations \$304/\$305(DBUFLO/HI) to see where the disk data is to be stored in RAM. This value, in lo/hi byte form, will be the start address of the data LOADED from the disk. Because the disk drive will only Read or Write a complete sector, only blocks of 128 bytes can be transferred at one time. For example, if DBUFLO/HI were given a value of 1536(\$600 hex), and the DCOMND were given a Read(LOAD) command, any data LOADED from the disk would be stored from 1536 to 1536 + 128(\$600 + \$80). However, when the DCOMND were given a Write(SAVE) command, the DOS would refer to DBUFLO/HI to see which part of RAM is to be stored on disk. For example, if the DBUFLO/HI were given a value of \$600, while DCOMND was in SAVE mode, RAM bytes 1536

# "A LIVING TAPESTRY . . ."



*"The world of Ultima III can only be compared to a living tapestry — complex and beautiful . . . This is the best fantasy game in computing. Indeed, it is one of the best fantasy worlds in which to live. Lord British is a veritable JRR Tolkien of the keyboard." — Popular Mechanics*

*"Exodus: Ultima III, with a superior plot to match its superior gaming system, is a great game. It upgrades the market; in several ways it sets new standards for fantasy gaming state of the art." — Softline*

*"Exodus: Ultima III is Lord British's magnum opus — so far. It's fun and exciting to play and constantly intriguing. And the ending is marvelously unexpected and not a bit disappointing — except that it is the ending, and as with a good book, you'll probably wish there were more." — Softalk*

Available on: Apple, Atari, Com64, IBM



1545 OSGOOD ST., #7 NORTH ANDOVER, MA 01845  
(617) 681-0609

Apple, Atari, Com64, and IBM are trademarks of Apple Inc., Atari Inc., Commodore Business Machines, and IBM, respectively.  
Ultima and Lord British are trademarks of Richard Garriott. Copyright 1984 by Origin Systems, Inc.



(\$600) to 1664 (\$680) would be stored on disk. Therefore, the location in free RAM where disk data is transferred(to and from), is left to the programmer's discretion. Addresses \$600 to \$6FF usually make an excellent transfer buffer. In fact, the LOAD and SAVE subroutines use the upper portion of these addresses for this purpose.(\$680-\$6FF)

The fourth step is to indicate to the DOS which sector to Read from or Write to. To do this, store the number of the sector in which you want to Read or Write into locations \$30A/30B(DAUX1/2). As a result, when the DCOMND is given the Read(LOAD) command, the DOS will examine locations \$30A/30B to determine which sector to LOAD data from. Conversely, when the DCOMND is given a Write(SAVE) command, the DOS will examine locations \$30A/30B to determine which sector to SAVE data to. The DAUX1/2 does much the same thing for the disk as the DBUFLO/HI does for the RAM.

The final step is to start the disk drive. This is simply accomplished by jumping to the Disk handler(\$E453). In the LOAD and SAVE routines this is done on line 220 with the JSR command.

```
00010 .LI OFF
00020 .OR$3A98 ;START ADDRESS
00022 *****
00024 *
00025 * LOAD ROUTINE /MACHINE CODE *
00027 *
00028 *****
00030 DUNIT .EQ $301 ;DRIVE NUMBER
00040 DCOMND .EQ $302 ;DISK OPERATION
00050 DBUFLO .EQ $304 ;SOURCE OF DATA
00060 DBUFHI .EQ $305 ;READ FROM DISK
00070 DAUX1 .EQ $30A ;DISK SECTOR TO
00080 DAUX2 .EQ $30B ;BE READ
00100 START LDA #1
00110 STA DUNIT ;SET UP DRIVE 1
00120 LDA #82 ;LOAD INDICATOR
00130 STA DCOMND ;DISK OPERATION
00140 LDA #128 ;BUFFER ADDRESS
00150 STA DBUFLO ;OF THE DISK
00160 LDA #6 ;SECTOR DATA
00170 STA DBUFHI
00180 LDA $600 ;SECTOR TO READ
00190 STA DAUX1 ;LO BYTE
00200 LDA $601
00210 STA DAUX2 ;HI BYTE
00220 JSR $E453 ;START DRIVE
00230 RTS ;RETURN TO BASIC
```

```
00010 .LI OFF
00020 .OR$3E80 ;START ADDRESS
00022 *****
00024 *
00025 * SAVE ROUTINE /MACHINE CODE *
00027 *
00028 *****
00030 DUNIT .EQ $301 ;DRIVE NUMBER
00040 DCOMND .EQ $302 ;DISK OPERATION
00050 DBUFLO .EQ $304 ;SOURCE OF DATA
00060 DBUFHI .EQ $305 ;WRITTEN TO DISK
```

```
00070 DAUX1 .EQ $30A ;DISK SECTOR
00080 DAUX2 .EQ $30B ;WRITTEN TO
00100 START LDA #1
00110 STA DUNIT ;SET UP DRIVE 1
00120 LDA #87 ;SAVE INDICATOR
00130 STA DCOMND ;DISK OPERATION
00140 LDA #128 ;ADDRESS OF DATA
00150 STA DBUFLO ;WRITTEN TO DISK
00160 LDA #6
00170 STA DBUFHI
00180 LDA $603 ;THE SECTOR NUMBE
R
00190 STA DAUX1 ;WHICH CONTAINS
00200 LDA $604 ;THE DATA
00210 STA DAUX2 ;HI BYTE
00220 JSR $E453 ;START DRIVE
00230 RTS ;RETURN TO BASIC
```

```
100 REM *****
105 REM *
110 REM * SECTOR EXAMINER *
120 REM *
122 REM *****
130 DIM CA$(2),K$(71),B$(113),CAR
$(16),NU(128),OLS(20)
140 GRAPHICS 1:SETCOLOR 2,16,1
150 POKE 752,1
160 POSITION 4,7
170 ? CHR$(125):? #6;"Please Wait"
180 POKE 752,0
190 GOSUB 1000
200 REM *
210 REM * DISK SUBROUTINE *
220 REM *
230 RESTORE 240:FOR Y1=1 TO 64:READ W:
K$(Y1,Y1)=CHR$(W):NEXT Y1
232 REM *
235 REM * LOAD DATA FROM THE DISK A51
*
238 REM *
240 DATA 104,104,104,104,141,1,6,162,8
2
250 DATA 201,83,144,2,162,87,142,0,6,1
69,6,141,5
260 DATA 3,169,128,141,4,3,169,1,141,1
,3,174
270 DATA 0,6,142,2,3,104,141,10,3,172,
1,6,140,11,3,32
280 DATA 83,228,160,0,140,213,0,172,3,
3,140,212,0,96
290 CAR$="0123456789ABCDEF"
300 OPEN #1,4,0,"K"
310 REM *
320 REM * MAIN DISPLAY
330 REM *
340 GRAPHICS 0
350 ? CHR$(125)
360 ? ," DISK EXAMINER"
370 ? :? :? "Enter the sector you want
to"
380 ? "examine. Put a '$' before any"
390 ? "hexadecimal entries."
400 ?
410 ? "for example:"
420 ? :? "$FA = Hexdecimal"
430 ? "250 = Decimal"
440 POSITION 2,20:? CHR$(156):? "Sect
or #";
450 INPUT B$
460 REM
470 REM * HEX NUMBER CONVERSION *
480 REM
490 IF B$(1,1)<>"$" THEN GOTO 590
500 B$=B$(2)
510 IF LEN(B$)>3 THEN 1590
520 IF LEN(B$)=2 THEN B$(3)=B$(2
):B$(2,2)=B$(1,1):B$(1,2)="0"
530 IF LEN(B$)=1 THEN B$(3)=B$(1
):B$(1,3)="00"
540 IF ASC(B$(1,1))>64 THEN B$(1,1
)=CHR$(ASC(B$(1,1))-7)
550 IF ASC(B$(2,2))>64 THEN B$(2,2
)=CHR$(ASC(B$(2,2))-7)
560 IF ASC(B$(3,3))>64 THEN B$(3,3
)=CHR$(ASC(B$(3,3))-7)
570 TE=(ASC(B$(3,3))-48)+16*(ASC(B$
$(2,2))-48)+256*(ASC(B$(1,1))-48)
580 GOTO 630
590 TRAP 620
600 TE=VAL(B$)
610 GOTO 580
620 GOSUB 1590:TE=-1:GOTO 580
630 OLS=B$
```

```

640 HAN=TE
650 REM
660 REM ** READ SECTOR **
670 REM
680 X=USR(ADR(KANS),82,HAN)
690 IF X=1 THEN 720
700 POSITION 2,19
710 ? "BAD SECTOR ";HAN
720 IF X=1 THEN GOSUB 770
730 GOTO 440
740 REM
750 REM ** DISPLAY SECTOR **
760 REM
770 TE=INT(HAN/256):? CHR$(125)
780 ? "SECTOR # = ";HAN;
790 ? " ($";
800 CSH=TE
810 TE=INT(TE/16)+1
820 CAQ$(1,1)=CAR$(TE,TE)
830 TE=(CSH-(TE-1)*16)+1
840 CAQ$(2,2)=CAR$(TE,TE)
850 TE=CSH: ? CAQ$;
860 TE=HAN-256*INT(HAN/256)
870 CSH=TE
880 TE=INT(TE/16)+1
890 CAQ$(1,1)=CAR$(TE,TE)
900 TE=(CSH-(TE-1)*16)+1
910 CAQ$(2,2)=CAR$(TE,TE)
920 TE=CSH: ? CAQ$;
930 ? " )";
940 X=USR(ADR(BRHS),1536+128)
950 OLS=BHAS
960 POSITION 2,20: ? CHR$(156);: ? "Sect
or # or Characters 'C'";:INPUT BHAS
970 IF BHAS="C" THEN ? "CALCULATING...
one moment":GOTO 1140
980 GOTO 490
990 REM
1000 REM ** DISPLAY MEM IN HEX **
1010 REM
1020 DIM BRHS(200)
1030 RESTORE 1040:FOR G1=1 TO 163:READ
W:BRHS(G1,G1)=CHR$(W):NEXT G1:RETURN
1032 REM *
1035 REM * RAM TO SCREEN SUBROUTINE A2
*
1038 REM *
1040 DATA 104,104,133,229,104,133,228,
169,0,72,104,72,16,7,169,155,32,164,24
0,104,96,169,155,32
1050 DATA 164,246,104,72,162,0,142,10,
6,160,0,224,0,240,72,169,32,32,164,246
,224,0,208,63
1060 DATA 74,74,74,74,201,10,48,2,105,
6,105,48,32,164,246,224,0,208,44,41,15
,201,10,48
1070 DATA 2,105,6,105,48,32,164,246,22
4,0,208,27,104,72,168,185,128,6,200,22
4,0,208,16,104
1080 DATA 162,5,142,10,6,24,105,1,72,4
1,7,208,24,240,155,238,10,6,174,10,6,2
24,1,240
1090 DATA 183,224,2,240,32,224,3,240,1
66,224,4,240,162,224,5,240,203,224,6,2
40,163,224,7,240
1100 DATA 195,224,8,240,174,224,9,240,
142,224,10,240,194,104,72,224,0,208,16
0
1110 REM
1120 REM * CONVERT HEX TO CHARACTERS *
1130 REM *
1140 BR=PEEK(88)+256*PEEK(89)
1150 E=3
1160 XC=0:W=1
1170 XR=BR+126
1180 P1=PEEK(XR)
1190 IF P1>30 THEN HB=P1-23:GOTO 1210
1200 HB=P1-16
1210 P=PEEK(XR+1)
1220 IF P>30 THEN LB=P-23:GOTO 1240
1230 LB=P-16
1240 NU(W)=HB*16+LB
1250 W=W+1
1260 XC=XC+1
1270 XR=XR+3
1280 IF W=128 THEN 1330
1290 IF XC>7 THEN XC=0:GOTO 1310
1300 GOTO 1180
1310 XR=XR+16
1320 GOTO 1180
1330 GRAPHICS 0
1340 Y=0
1350 Y1=3
1360 POSITION 5,1: ? "SECTOR # ";OLS;"

```

```

(CHARACTERS)"
1370 FOR X=1 TO 128
1380 Y=Y+1
1390 IF NU(X)>251 THEN NU(X)=251
1400 IF Y>7 THEN POSITION Y*3+5,Y1: ? C
HRS(NU(X));:Y=0:Y1=Y1+1:GOTO 1420
1410 POSITION Y*3+5,Y1: ? CHR$(NU(X));
1420 NEXT X
1430 POSITION 3,3: ? "00"
1440 POSITION 3,4: ? "08"
1450 Y=3
1460 FOR X=10 TO 70 STEP 10
1470 Y=Y+2
1480 POSITION 3,Y: ? X
1490 NEXT X
1500 Y=4
1510 FOR X=18 TO 78 STEP 10
1520 Y=Y+2
1530 POSITION 3,Y: ? X
1540 NEXT X
1550 GOTO 440
1560 REM
1570 REM * INCORRECT *
1580 REM
1590 POSITION 2,19
1600 ? " INCORRECT"
1610 GOTO 440

```

```

90 REM * CHECK DATA *
100 DATA 8598,390,225,838,222,394,296,
9,923,150,332,925,950,635,857,637,175,
640
235 DATA 9200,508,646,111,571,371,771,
749,651,131,637,529,639,11,924,511,475
,965
390 DATA 6539,487,430,664,25,528,869,1
1,601,274,603,77,6,742,79,82,527,534
560 DATA 10819,541,969,842,840,98,840,
888,831,770,602,615,604,784,239,199,50
3,654
730 DATA 10254,838,602,865,604,937,498
,677,775,265,555,469,559,241,753,782,2
72,562
900 DATA 8784,467,557,239,547,825,836,
95,43,850,609,105,641,115,817,687,658,
693
1040 DATA 10940,900,565,429,628,311,71
2,986,642,560,686,809,567,906,963,123,
295,858
1210 DATA 8886,160,198,816,246,714,851
,884,434,69,930,931,932,59,585,638,206
,233
1380 DATA 10127,722,25,480,388,825,511
,521,590,706,723,478,832,587,718,719,4
74,828
1550 DATA 4137,887,651,481,653,255,326
,884

```

```

00010 .LI OFF
00020 .OR $4000 ;OBJECT CODE
00030 *****
00040 * *
00050 * AS1 *
00060 * LOAD DATA FROM THE DISK *
00070 * *
00080 *****
00100 * MACHINE CODE 1
00110 PLA
00120 START PLA ;PULL BYTES OF THE
00130 PLA ;STACK
00140 PLA
00150 PLA
00160 PHA
00170 STA $601
00180 LDX #82 ;DETERMINE WHICH
00190 CMP #83 ;DISK OPERATION
00200 BCC L1 ;TO BE PERFORMED
00210 LDX #87 ;WRITE(verify)
00220 LI STX $600
00230 LDA #6
00240 STA $305 ;BUFFER ADDRESS
00250 LDA #128 ;OF THE DISK SECTOR
00260 STA $304 ;DATA
00270 LDA #1 ;DEVICE SERIAL BUS
ID
00280 STA $301 ;SET TO DRIVE 1
00290 LDX $600
00300 STX $302 ;SET DISK OPERATION
00310 PLA ;INPUT DISK SECTOR
00320 STA $30A ;NUMBER
00330 LDY $601
00340 STY $30B
00350 JSR $D453
00360 LDY #0

```

```

00370 STY $D5 ;FLOATING POINT
00380 LDY $303 ;REGISTER
00390 STY $D4
00400 RTS ;RETURN TO BASIC
00010 .LI OFF
00020 .OR $3A98 ;STORE OBJECT CODE
00100 *****
00110 *
00120 * AS2 *
00130 * RAM TO SCREEN SUBROUTINE *
00140 *
00150 *****
00160 START PLA ;PULL BYTE OF STACK
00170 PLA
00180 STA $E5 ;STORE 'A' IN THE
00190 PLA ;FLOATING POINT
00200 STA $E4 ;REGISTER 1
00210 LDA #0
00220 PHA ;PUSH 'A' ON THE ST
ACK
00230 L8 PLA
00240 PHA
00250 BPL L1
00260 LDA #155 ;MOVE THE CURSOR
00270 JSR $F6A4 ;ONE SPACE
00280 PLA
00290 RTS ;RETURN TO BASIC
00300 *
00310 L1 LDA #155 ;MOVE THE CURSOR
00320 JSR $F6A4 ;ONE SPACE
00330 PLA
00340 PHA
00350 LDX #0 ;SET OPERATION
00360 STX $60A ;COUNTER
00370 LDY #0 ;SET MEMORY COUNTER
00380 CPX #0
00390 BEQ CON ;GOTO CONTROL SECTI
ON
00400 *
00410 * SECTION 1
00420 *
00430 S1 LDA #32 ;MOVE THE CURSOR
00440 JSR $F6A4 ;ONE SPACE
00450 CPX #0
00460 BNE CON ;RETURN TO CONTROL
SECTION
00470 *
00480 * SECTION 2
00490 *
00500 S2 LSR ;DETERMINE THE FIRS
T
00510 LSR ;DIGIT OF THE
00520 LSR ;BYTE NUMBER
00530 LSR
00540 CMP #10
00550 BMI L4
00560 ADC #6
00570 L4 ADC #48
00580 JSR $F6A4 ;DISPLAY IT
00590 CPX #0
00600 BNE CON ;RETURN TO CONTROL
SECTION
00610 *
00620 * SECTION 3
00630 *
00640 S3 AND #15 ;DETERMINE THE SECO
ND
00650 CMP #10 ;DIGIT OF THE
00660 BMI L5 ;BYTE NUMBER
00670 ADC #6
00680 L5 ADC #48
00690 JSR $F6A4 ;DISPLAY IT
00700 CPX #0
00710 BNE CON ;RETURN TO CONTROL
SECTION
00720 *
00730 * SECTION 4
00740 *
00750 S4 PLA
00760 PHA
00770 TAY ;LOAD ONE BYTE OF
00780 LDA $680,Y ;DATA FOR THE MEMOR
Y
00790 INY ;INCREMENT MEMORY C
OUNTER
00800 CPX #0
00810 BNE CON ;RETURN TO CONTROL
SECTION
00820 *
00830 * ENDING
00840 *
00850 EN PLA
00860 LDX #5 ;RESET OPERATION CO
UNTER
00870 STX $60A
00880 CLC
00890 ADC #1
00900 PHA
00910 AND #7
00920 BNE L7 ;NEXT NUMBER
00930 BEQ L8 ;NEXT LINE(or end)
00970 CON INC $60A ;INCREMENT OPERATIO
N COUNTER
00980 LDX $60A ;LOAD OPERATION COU
TER
00990 CPX #1
01000 BEQ S2 ;GOTO SECTION 2
01010 CPX #2
01020 BEQ S31 ;GOTO SECTION 3 SUB
01030 CPX #3
01040 BEQ S1 ;GOTO SECTION 1
01050 CPX #4
01060 BEQ S1 ;GOTO SECTION 1
01070 L7 CPX #5
01080 BEQ S4 ;GOTO SECTION 4
01090 CPX #6
01100 BEQ S2 ;GOTO SECTION 2
01110 CPX #7
01120 BEQ S4 ;GOTO SECTION 4
01130 CPX #8
01140 BEQ S3 ;GOTO SECTION 3
01150 CPX #9
01160 BEQ S1 ;GOTO SECTION 1
01170 CPX #10
01180 BEQ EN ;GOTO END SECTION
01190 *
01200 * SECTION 3 SUB 1
01210 *
01220 S31 PLA
01230 PHA
01240 CPX #0
01250 BNE S3 ;TO SECTION 3
5 REM *****
6 REM *
7 REM * FILE FINDER *
8 REM *
9 REM *****
10 GRAPHICS 1
12 DIM A$(5),PH(5),MSH(5),NVS(5)
15 POKE 752,1
20 SETCOLOR 2,16,1
25 POSITION 0,8
30 ? #6;"LOADING MACHINE CODE"
49 REM * STORE SAVE ROUTINE *
50 FOR X=0 TO 35
52 READ D
55 POKE 16000+X,D
60 NEXT X
69 REM * STORE LOAD ROUTINE *
70 FOR X=0 TO 35
72 READ D
75 POKE 15000+X,D
80 NEXT X
100 POKE 14999,104
110 POKE 15999,104
120 REM *
122 REM * WHAT SECTOR TO MODIFY *
125 REM *
130 GRAPHICS 0:POKE 752,1
132 POSITION 15,1:?"FILE FINDER"
135 POSITION 2,4
140 ? "What sector to modify(dec)":FOR
X=1 TO 50:NEXT X:POKE 764,255:INPUT M
5
142 IF M5<1 OR M5>719 THEN ? "(1-719)"
:GOTO 140
145 MSC=M5/256
150 MSM=(MSC-INT(MSC))*16
155 MSH(1)=INT(MSC)
160 MSM1=INT((MSM-INT(MSM))*16+0.5)
165 MSH(2)=INT(MSM)*16+MSM1
210 POKE 1536,MSH(2)
220 POKE 1537,MSH(1)
230 POKE 1539,MSH(2)
240 POKE 1540,MSH(1)
250 X=USR(14999)
260 GOTO 1000
265 REM *
267 REM * DISPLAY BYTE NUMBERS *
268 REM *
270 GRAPHICS 0
271 POKE 752,1
272 POSITION 0,1:?"BYTE NO. SECT
OR: ";MS:POSITION 2,2:?"dec
HEX"
273 FOR X=0 TO 15
274 POSITION 2,X+3
275 ? X#8
284 NEXT X
298 REM *
299 REM * DISPLAY TO SCREEN *
305 REM *
310 FOR X=1664 TO 1792 STEP 8

```



```

320 FOR Y=1 TO 8
330 PD=PEEK(X+Y-1)
340 PHC=PD/16
350 PH(2)=INT((PHC-INT(PHC))*16+0.5)
360 PH(1)=INT(PHC)
365 FOR WA=1 TO 2
370 POSITION Y*3+WA+4,INT((X-1664)/8)+
3
380 IF PH(WA)>9 THEN 400
390 ? PH(WA)
395 GOTO 470
400 IF PH(WA)=10 THEN AS="A"
410 IF PH(WA)=11 THEN AS="B"
420 IF PH(WA)=12 THEN AS="C"
430 IF PH(WA)=13 THEN AS="D"
440 IF PH(WA)=14 THEN AS="E"
450 IF PH(WA)=15 THEN AS="F"
460 ? AS
470 NEXT WA
475 IF X+Y>1791 THEN 490
480 NEXT Y
485 NEXT X
490 ? :? "Press the SPACE BAR to conti
nue"
500 IF PEEK(764)<>33 THEN 500
510 GOTO 1000
690 REM *
692 REM * DISPLAY CHARACTERS *
695 REM *
700 GRAPHICS 0
702 POKE 752,1
704 POSITION 0,1:? "BYTE NO.      SECTOR
: ";MS:POSITION 2,2:? "dec      CHA
R"
705 FOR X=0 TO 15
710 POSITION 2,X+3
725 ? X*8
735 NEXT X
740 FOR X=1664 TO 1792 STEP 8
750 FOR Y=1 TO 8
760 PD=PEEK(X+Y-1)
770 POSITION Y*3+4,INT((X-1664)/8)+3
780 ? CHR$(PD)
800 IF X+Y>1791 THEN 830
810 NEXT Y
820 NEXT X
830 ? :? "Press the SPACE BAR to conti
nue"
840 IF PEEK(764)<>33 THEN 840
850 GOTO 1000
990 REM *
994 REM * MAIN LOOP *
995 REM *
1000 GRAPHICS 0
1005 POKE 752,1
1010 POSITION 13,1
1015 ? "FILE FINDER"
1018 POSITION 7,3
1020 ? "What do you want to do?"
1025 POSITION 1,6
1030 ? "For character interpretation p
ress 'C'"
1040 POSITION 1,7
1050 ? "For HEX interpretation press
'H'"
1054 POSITION 1,8
1055 ? "To select a new sector press
'M'"
1060 POSITION 2,10
1070 ? "To make modification press 'M'
"
1080 POSITION 5,12
1090 ? "INPUT?"
1095 POKE 764,255
1100 IF PEEK(764)=18 THEN 700
1110 IF PEEK(764)=57 THEN 270
1120 IF PEEK(764)=37 THEN 2000
1125 IF PEEK(764)=35 THEN 130
1130 GOTO 1100
2000 POSITION 0,14
2020 ? "Modify what byte number?(answe
r in dec.)"
2022 POKE 764,255
2025 FOR X=1 TO 50:NEXT X
2030 POSITION 5,15
2040 INPUT BN
2050 POSITION 0,16
2060 ? "Change to what value(DEC. OR $
HEX)"
2070 POSITION 5,17
2080 INPUT NV$
2085 IF NV$(1,1)="$" THEN DH=16:GOTO 4
000
2087 GOTO 3800
2090 POKE 1664+BN,NV

```

```

2100 POSITION 2,18
2110 ? "OK"
2120 X=USR(15999)
2130 GOTO 130
3800 IF LEN(NV$)=3 THEN LN=100
3810 IF LEN(NV$)=2 THEN LN=10:GOTO 391
0
3820 IF LEN(NV$)=1 THEN LN=1:GOTO 3920
3900 N100=(ASC(NV$(3,3))-48)*INT(LN/10
0)
3910 N10=(ASC(NV$(2,2))-48)*INT(LN/10)
3920 N1=(ASC(NV$(1,1))-48)*LN
3930 NV=N100+N10+N1
3940 IF NV<0 OR NV>255 THEN 2070
3950 GOTO 2090
3990 REM *
3995 REM * CONVERT TO DECIMAL *
3997 REM *
4000 IF ASC(NV$(2,2))>64 THEN HB=ASC(N
V$(2,2))-55:GOTO 4020
4010 HB=ASC(NV$(2,2))-48
4015 IF LEN(NV$)<3 THEN LB=HB:HB=0:GOT
O 4040
4020 IF ASC(NV$(3,3))>64 THEN LB=ASC(N
V$(3,3))-55:GOTO 4040
4030 LB=ASC(NV$(3,3))-48
4040 NV=HB*DH+LB
4050 GOTO 2090
7999 REM * MACHINE CODE (SAVE routine)
*
8000 DATA 169,1,141,1,3,169,87,141,2,3
,169,128,141,4,3,169,6,141,5,3,173,3,6
,141
8010 DATA 10,3,173,4,6,141,11,3,32,83,
228,96
9999 REM * MACHINE CODE (LOAD routine)
*
10000 DATA 169,1,141,1,3,169,82,141,2,
3,169,128,141,4,3,169,6,141,5,3,173,0,
6,141
10010 DATA 10,3,173,1,6,141,11,3,32,83
,228,96

```

**Non-Standard Magic!**  
Software Design  
P.O. BOX 48  
GIRARD, OHIO 44420

**\*LISTER PLUS\***

The COMPLETE printer utility program for ATARI computers. With \*Lister Plus\*, a good quality interface, and just about ANY graphics capable printer, your printer will print ALL Characters and ! It will dump screens that have been created with MICROPAINTER, \*PICTURE PLUS\*, Movie Maker, Fun With Art, Keala Pad, or any graphics tool that saves straight binary graphics seven + or graphics eight screen.

48K DISK.....ONLY \$19.95

**\*PICTURE PLUS\***

An easy to use graphics utility designed to supplement your present graphics software, such as Keala Pad, Atari Touch-Tablet, B-Graph and Micropainter, and no special tools are needed. Once you have loaded a graphics file you can overlay text in FIVE different heights in four colors using the standard ATARI character set, the \*PICTURE PLUS\* resident custom character set or one of your own favorites! \*Picture Plus\* also includes a resident disk utility package (menu driven) with full features that supports two drives so you'll never be left with a screen you can't save.

48K DISK.....ONLY \$29.95

**\*JOURNAL PLUS\***

A date oriented text file creator that will enable you to easily keep that diary you've been thinking with your own personal ID code. Or, choose a universal code for family records or a "baby book" that anyone can access. You can page forward or backward through your files, or \*Journal Plus\* will search for any 15 character string you desire through an entire disk!

48K DISK.....ONLY \$19.95

**Non-Standard Magic!**

# Game Reviews

## F-15 STRIKE EAGLE Reviewed by Peter Ellison

F-15 Strike Eagle is a game that simulates flying a fighter through seven different scenarios. Each Mission is based on an actual historical event, adding realism to the simulation.

I'm not a real fan of strategic, or war games, so, when I first looked at the package and saw the thirty-six page manual, I thought, "This game is not for me!" When I read in the manual that it required one to two joysticks, I knew that I'd better have a closer look.

After booting up the system and pushing start, I saw that there was something different about this game. The screen read 'Authenticate and Enter Code.' So, I looked through the manual to find a part that read 'Top Secret!,' and typed in the secret authorization code.

This game, written by Sid Meier, author of Hellcat Ace, and Major Bill Stealy, President of MicroProse, and fighter aircraft pilot for fourteen years, brings the F-15 Eagle home to you. I never thought I'd be able to feel what an F-15 fighter pilot might experience. With this excellent simulation, I can.

The screen shows the cockpit of an F-15 Eagle, including everything from the Heads-Up-Display (HUD), to the Instrument Panel. By pressing the spacebar, you're able to look out the rear of the aircraft. The rear view shows only the sky, ground or sea surface, and other aircraft or missiles. The HUD includes: Airspeed, Altitude, Aircraft ground reticle, air-to-air reticle, air-to-ground reticle and pitch lines, target designator box, missile designator box, and steering cue.

Besides having all of the aircraft systems information showing on the HUD, there are also a number of messages that appear from time to time. When you are firing your gun it replies by telling you how many rounds remain. 'Missile Armed' appears when arming either a short, or a long-range missile. Other messages include 'bomb armed,' 'enemy plane hit,' 'bombs released,' 'bombs missed,' and 'target hit.' As well as messages that appear on the screen, warnings are flashed on. These warnings are: 'Alert: Sam Launched,' 'Damage Warning,' 'Alert: Air Missile.'

Also, when using defensive measures, messages appear referring to different actions. These messages indicate whether you're using long, medium, or short range radar, ECM Jamming, and Flare Released.

Up Front Controls (UFC) are operated by either the keyboard or second joystick. I found the keyboard a

lot easier and faster to operate while flying, since I did not have to remember the positions of the second joystick. These UFC's include activate gun mode, activate short or medium range missile, arm bomb, throttle, afterburner, and many more commands. When first playing this game, I thought that I would never be able to remember all the commands, but after only a time or two, I had them all memorized.

Below the HUD, and to the right, is an outline of your Eagle, showing the Weapon Status Display. This shows how many bombs, short or medium missiles, and flares there are, and the fuel status. At a quick glance, it is easy to get the status of your weaponry.

One panel to the left is your radar display. The radar can be set at either long, medium, or short range. When I'm flying, I usually keep it at medium, switching to long for a quick scan, or to short when in close combat. The radar is interesting to watch after you have released a short or medium range missile to track down another fighter, or when you have released a flare to mislead a missile which has been fired toward you.



The final panel to the left shows the horizontal situation display (HSD). This panel marks out the different Sam missile sites, airfields, primary targets, and the F-15 base. A navigation cursor can be moved around this screen by the use of the control arrows on the keyboard. To designate a target, you move the cursor to the top of the location you want to bomb. The steering cue, which is located on your HUD, will now be the direction in which you want to fly. If you continue to fly in this direction, you will eventually reach your destination. All this equipment makes air combat exciting.

There are seven missions, as explained in the first paragraph: Libya, August 19, 1981, Egypt, October 6, 1973, Haiphong, April 15, 1972, Syria, March 12, 1984, Hanoi, May 10, 1972, Iraq, June 7, 1981, and Persian Gulf, June 5, 1984. Each mission consists of



bombing airfields, SAM's, and one to three primary targets. Once all the primary targets are destroyed, and your fighter returns to base, the mission is over. You are able to destroy a few SAM's or airfields, and return to base to refuel, make repairs, and replenish weapons, and then to continue the mission. The more things you destroy, the more points you receive; but to continue onto the next mission, you must make it back to your base, or bail out and be rescued.

The aircraft which you'll encounter range from a Mikoyan/Gurevich Mig-21 to a Sukhoi Su-22. SAM's fire SA-2: Ground to air missiles (radar guided), SA-4: Ground radar guided and semi-active radar homing in terminal phase, and SA-7: Hand-held, infra-red missiles.

The simulation has four skill levels: Arcade, Rookie, Pilot, and Ace. The arcade level is more for the beginner because the aircraft does not roll, and it enables one to become familiar with the instrumentation and weapons. Rookie is what I usually play (at this time), but when I want a real challenge, I select ACE. With that, I don't last too long, but it's fun trying.

The instructions or flight operations manual is one of the best documentations of any game or simulation that I've played. Thirty six informative pages concerning F-15 specifications, weapons, cockpit layout, diagrams, and aerodynamics, makes this manual, in itself, a learning experience. I was never interested in fighter aircraft until I read this manual. I rarely enjoy reading documentation, but for this game, it was worth it.

F-15 Strike Eagle is one of the best games that I've seen for a long time. This game involves strategy, good reflexes, and skill, which makes it challenging for anyone, whether an arcade or strategy game player. This simulation combines the two for an entertaining game that will keep you busy for years to come. To be able to fly a \$20,000,000 aircraft for a suggested retail price of \$34.95 for the game, is a steal.

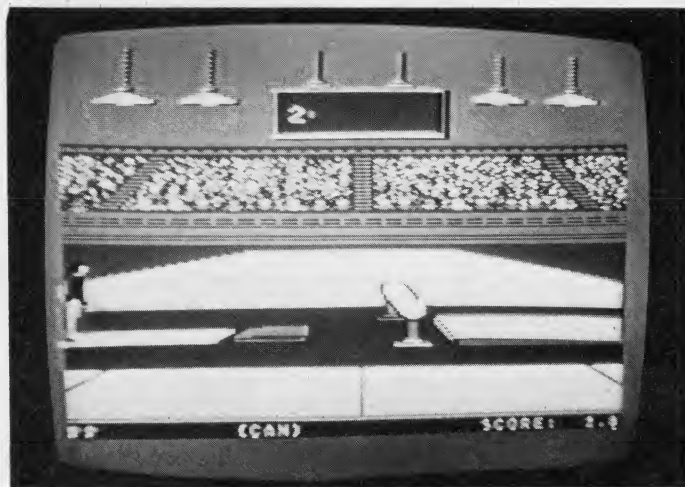
**F-15 STRIKE EAGLE**  
MicroProse Software  
10616 Beaver Dam Road  
Hunt Valley, MD 21030

Ratings  
Playability: 9.5  
Challenge: 10  
Graphics: 9.2  
Sound: 8.0  
Documentation: 10  
Overall: 9.5

## SUMMER GAMES Reviewed by Gabe Torok

Here I am, the closest I get to a sport, any sport, is a beer and television, and all of a sudden I am required to participate in the Olympics. No, not the Special Olympics! I don't consider my spare tire a handicap, rather, extra energy stored studiously for moments just like this.

The Olympics in which I participated several times was the 1984 Summer Games, produced for my frustration by Epyx. I say frustration because I performed about the same as if I had been there in person. Perhaps my lack of T-shirt sales would have added to my losses, so I may consider losing in the privacy of my own home a blessing. Who wrote this program, anyway? Just how many Olympians are computer programmers, that can produce such realism, with such fantastic graphics?



With nine countries to choose from, I may have kept picking the losers. After the anthem, the opening ceremonies, and a careless look at the instructions, I was ready. Or so I thought. The first event was the pole vault. I'm not good with heights, but I gripped my pole and gave it a good run. Got the pole firmly set into the vault box, — and forgot to let go. FAULT! They gave me a second chance. Better. I let go of the pole, perhaps some would say a little too soon. FAULT! They were really nice about the whole thing. One last chance, and I flubbed it. Perfect vault, perfect release, graceful drop, hard landing. I missed the mat. Not even a bronze for this one, I thought. Wrong! As I was the only one competing, I got the gold!

The next event was much better. Diving from a sofa is not at all difficult. I performed like a veteran, full tuck position, layout or pike, you name it. I felt like Bo Derek when I saw all the scores come in. I did so well, they let me do several types of dives until my score got back to a normal low. But I still got the



gold.

The 4x400 Meter Relay played havoc with my joystick but I managed to keep up with the computer pacer. It really felt good to win again.

After the exertion of the relay, a cooling pint of beer, a bag of pretzels, and a life-giving cigarette, I was ready for the 100 yard dash. Piece of cake after my previous success, and I chalked up another gold with little trouble.

Gymnastics? Me? You gotta be kidding! I sat up straight, mustered all my concentration, pressed the start button, sprang, and missed the horse. (No, really, that's what they called it. The only resemblance I could see were the four legs and the saddle on which I caught myself on the second try.) The third try revealed to me why the name 'horse.' I made a perfect vault, flipped, turned in mid air, came down almost feet first. Where I landed reminded me of my last pony ride, and the pain that followed. (In those days they used to call me 'tall in the saddle' until my blister broke.)

Now you're cookin'. Swimming is what I do best. The water holds all my parts in a streamlined bouyancy that allows me to imitate a veritable torpedo. No-one ever told me I had to push a joystick button to use my notorious power stroke? I was much better on the second lap, and almost great on the third. Thus, my sixth gold. The seventh came on the 100 meter freestyle, by which time I mastered the intricacies of pushbutton control.

Skeet Shooting? Terrific!!!! Finally, I can rest a bit after the strenuous events of the past half hour! (They don't even give you time to change.) There I was in this half circle somewhere, looking for the skeets. Nothing happened. Moving the joystick revealed the location of my shotgun, and reading the instructions revealed the release mechanism for the doves (or pigeons, depending on where you live). I hit the button, releasing the pigeon, aimed, pressed the button again, but the little blighter was too fast for me. I tried again, but they're sneaky! They came at me from a different direction, and caught me unprepared. This time...I thought; I quickly pressed the button, somehow knowing they would come at me from the opposite side. I aimed and fired, creating hole number one in a newly proposed golf course. Mustering all my concentration, and getting the lay of things, I managed to end the sorrowful life of the next four pigeons. Surely there's a cleaner way to win a gold?

Summer Games

EPYX Inc.

Playability:9

Exercise value: 10

Graphics: 10

Sound: 8

Objections: Having to turn the disk back and forth through the first few events.

## Archon II Reviewed by Gavin Bamber

Clear your mind of all thoughts. Now, picture a barren island upon which there are two citadels. Listen to the gentle lapping of waves against the shore. Smell the fresh air. Look! Beyond the water is a sea of flames!

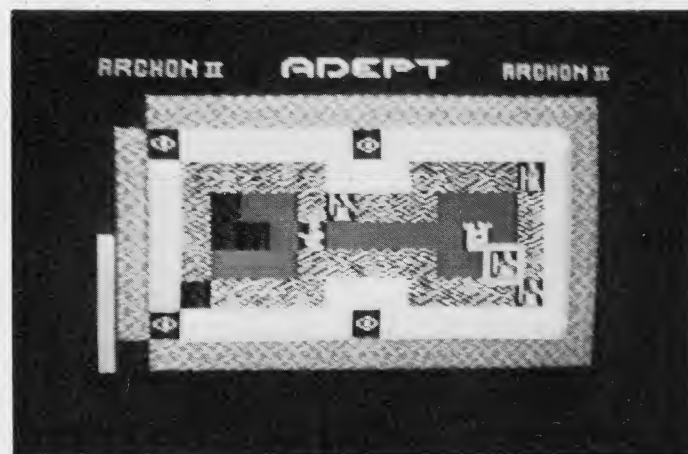
This is the magic realm where the Master of Order and the Mistress of Chaos fight for the supremacy of the world.

Attending these warlords are a few adepts, one for each of the classical elements composing this zone of magic: Air, water, earth, and fire. In turn, the adepts have assistants, four types of elements, and four forms of demons, which can be conjured up at a moment's notice.

Magic reigns supreme here as Order and Chaos vie for the constantly moving power squares. Spells are cast: Summon (to bring in a demon or elemental), heal, weaken, imprison, release, banish, and the ultimate: Apocalypse.

Watch as Chaos summons an elemental, the Siren, and moves it to a square occupied by the enemy Kraken. A battleground containing pools of water appears as the fighters assess each other's supply of energy. The Siren sings a sweet but deadly song which causes the slow-firing Kraken to lose strength. Quickly the Kraken generates a tidal wave across the water, and knocks the Siren into oblivion.

And so goes the game as each duel is fought in battlefields of rock, fire, air, or water. Spells are cast, and power squares conquered until it is time for the



final battle.

This apocalyptic battle is the beginning of the end as the Master of Order and Mistress of Chaos fight to the death in the void where one's strengths and

weaknesses depend on the amounts of energy, icons, and adepts left intact.

To return to the real world: "Archon II: Adept" is a much improved upon version of Archon. The additional spells, plus the shifting power squares, add a greater challenge to the players who are well acquainted with Archon.

One of the best features of this game is that there are three skill levels, so that one's capabilities may be matched pretty evenly with those of an opponent, whether human or computer. Beware though, the computer becomes trickier as you progress; the Adept is adaptable.

It is unfortunate that the instruction manual is rather short and to the point. In my view, Electronic Arts should have explained the situation in more detail, and the backgrounds of the icons were too brief. But, I suppose, something should be left up to the imagination of the players.

Archon was considered by many to be the best Atari game of 1983, and Archon II: Adept is sure to have that distinction in 1984.

It is equally remarkable that as a sequel to Archon, the Adept is not a copy of the original. It can be enjoyed by those who have Archon, as I do, as well as by those who don't, because it is quite different.

Overall, the game is great fun and challenging, with a good variety of playing pieces and screens. The graphics and sound effects for each element and icon are quite unique.

#### Ratings

Graphics: 10

Challenge: 9

Sound: 10

Documentation: 8.5

Holds Interest: 10

Overall Rating: 9.5

#### ARCHON II: ADEPT

Electronic Arts

2755 Campus Drive  
San Mateo, CA 94403

### Programmer's Wanted!!

Young or old, beginners or advanced, if you think your program is good send it in to us for evaluation. We consider every submission very carefully. So send in your programs today and join the ROM staff!

Send them to:

ROM Magazine

c/o Submissions

P.O. Box 252

Maple Ridge, B.C.

V2X 7G1

# EXPAND YOUR UNIVERSE

Alpha Systems proves again that excellent software doesn't have to cost a lot.  
All prices in American funds



## MAGNIPRINT

Prints your Atari® Graphics Screens like you've never seen before! Even prints GTIA modes with 16 shades. Prints various sizes from 1/8th page to GIANT Wall size posters. Allows you to enlarge and print any portion of the screen. Works with standard paper and Epson, NEC, C. Itoh or Gemini printers. Prints vertically or horizontally. Special feature lets you modify pictures on the screen. Prints your own screens or those from Graphics Master, Micropainter, Koala Pad, Atari® Touch Tablet, Fun with Art, PAINT, BiGraph, and others.  
FREE: With any Magniprint order-PRINTALL. Allows you to print your programs or files just as they appear on the screen. Clearly prints all graphics symbols, even INVERSE and control characters.

**Only \$20.95**

Print Giant Posters - Up to 6 feet

THE ULTIMATE GRAPHICS PRINTING PACKAGE FOR ATARI COMPUTERS. NO OTHER PROGRAMS CAN DO ALL THIS.

### THE Scalyzer

AT LAST A UTILITY THAT DOES IT ALL! Scans & Analyzes ALL Atari® programs. Works on programs stored on Disk, Cartridge or directly from memory. Converts complex ANY Atari® BASIC program into readable assembler. Transforms ANY Atari® BASIC program into

Instable, modifiable BASIC. Changes a 4, 8 or 16K cartridge into a binary load file and source file that you can view & change using regular Atari® assembler. Clearly shows protection techniques such as BAD SECTORS, BAD DATA MARKS, DUPLICATE SECTORS and FORCED CRC ERRORS. Even finds and displays hidden directories. No other program can do this!

**\$29.95**

Complete with instructions in theory and use



Actually done with Magniprint



**ATARI SOFTWARE PROTECTION TECHNIQUES**  
GEORGE MORRISON

Top Selling Book (over 100 pages)  
"ATARI SOFTWARE PROTECTION TECHNIQUES"  
Thoroughly explains how advanced software pirates copy programs, and how you can protect your programs. Complete with disk of protection programs.  
**\$24.95**

### KEYBOARD CUSTOMIZER

Customizes your Atari® to transform it into one of the most powerful program development tools ever. Allows you to alter functions of your keyboard to fit your personal needs. Allows you to give multiple commands that will execute automatically on systems start up or whenever you wish. Makes the computer seem to program itself. Can generate common program lines or statements from a single keystroke, greatly reducing typing time. Imagine hitting one key (or combination of keys) to generate any statement of your choice instantly on the screen! Lets cursor move 50% faster. Works perfectly with Basic, Assembler, Pilot, or all by itself. This 100% machine language program was developed by a large scale systems programmer for his own use, but is now available to everyone. Increases programming efficiency. Reduces keying errors. Easy enough for a beginner.  
**\$16.95**



### the IMPERSONATOR

Create normally running back-up copies of your cartridges. Yes, for only \$29.95 you can have working copies of all your 4K, 8K, or 16K game cartridges for Atari® computers. Special software you receive will allow you to save the data from a cartridge to an ordinary disk file. This disk file will run just like the original cartridge when used with The Impersonator. Now you can put all your real cartridges away for safe keeping and use The Impersonator for everything. Each disk can hold 5 or more cartridges.  
NOTE: This product is intended for use as a back-up tool for your own cartridges. Alpha Systems does not condone copying borrowed or rented cartridges.  
**ONLY \$29.95**



### Cassette Operating System (C.O.S.)

The only cassette program you'll ever need. COPIES: C.O.S. Copies all Atari® cassettes - Copies disk files to cassettes - Copies single boot cassettes to disk - Stores any cassette program to disk for safe keeping. DISPLAYS: Displays any cassette program in hex, ascii, or converts it to a readable assembler language file. (Compatible with Atari® assembler cartridge) ALTERS: Modifies the size, contents or combine and dissect whole programs. Modified files can be saved to disk or multi-stage cassette.  
All this on one disk for only \$24.95  
Includes complete cassette tutorial. Including adding music to your cassette.



**ALPHA SYSTEMS**

MAIL TO: Alpha Systems/4435 Maplepark Rd./Stow, OH 44224  
Send check or money order. Include \$2.00 shp. & hdlg.  
All prices in American funds  
CALL: 216-374-7469 to charge to MasterCard or VISA

BONUS: Order any 3 programs & get FREE Deluxe Space Games (3 games on a disc)

All for your Atari Computers - Disk drive and 48K required  
Atari is a registered trademark of Atari Corporation

# Sparta DOS & US Doubler

Reviewed by Peter Ellison

ICD has done something that I didn't think was possible. They have created a DOS for the Atari (SpartaDOS) that is the best that I have ever used. Besides doing that, they have created two chips that will make your 1050 disk drive store almost twice as much data on a single disk and run much faster (Ultra Speed Doubler).

I'll begin by trying to review a product that is so revolutionary it adds new life to the Atari. Little did I know that SpartaDOS would change my Atari's whole concept of what a Disk Operating System is.

After booting up the disk, I was surprised that the computer asked me to enter the date and time. Right from the beginning I knew this DOS was very different. The time was displayed in the upper right-hand corner of the screen. In the center, the date and day were displayed. This autorun file was created with the file called STARTUP.BAT, which exists on the master disk. For each disk, you can create your own STARTUP.BAT file, which will automatically execute upon bootup. A batch file contains executable DOS instructions that can be created with a word processing program, or with the Screen Editor using the COPY command.

Since SpartaDOS has so much to offer, I'll just give a brief overview of the goals that ICD set for themselves, and reached. SpartaDOS, which is memory resident, is completely compatible with all software and systems. It supports all densities and switch configurations intelligently. It is able to handle hard disk drives when they become available. It will allow file storage limited by only the capacity of the media. It allows time/date stamping of files in the directories. MEMLO is below \$2000, and supports batch file processing. It supports full redirection of I/O and relative file access. It is command driven for entry speed and easy expansion. It has provisions for UltraSpeed I/O.

The final goal was to make it disk compatible with DOS 2. Almost immediately, it was obvious ICD couldn't support all these features, and still maintain 'disk compatibility.' Although the two are not compatible, Mike Gustafson, the author, was able to develop a copy program that would transfer both to and from SpartaDOS with automatic configuration; thus, SPCOPY was born.

SpartaDOS contains both Internal and External commands, unlike that of its predecessor Atari DOS 2. Internal commands are memory resident, and

include: APPEND, Batch Processing, BUFS, CAR, COPY (providing Page 6 is not used, otherwise it's external), CREDIR, CWD, DELDIR, DIR, ERASE, LOAD, PRINT, RENAME, RUN, SAVE, and TYPE. The external commands include: AT—RS232, CLS, COPY (if using Page 6), DIS—BAT, DUPDSK, FORMAT, INIT, MEMLO, PAUSE, PORT, RS232, SET, SPCOPY, TIME, TREE, and UNERASE. For you to be able to use the external commands, files of the same name must reside on the disk from where they are called. So, to provide more disk storage space, copy only the command files that you will be using to each disk, and keep the utilities disk near by.

The commands included, that are not available in Atari DOS 2.0S are: Batch Files, BUFS, CREDIR, CLS, CWD, DELDIR, DIS—BAT, INIT, MEMLO, PAUSE, PORT, SET, TIME, TREE, and UNERASE. I'll give a brief explanation of each of the commands that I haven't mentioned yet. The first, BUFS, is used to set or check the number of buffers currently in use. CREDIR creates a subdirectory on the specified disk. This is great for keeping track of related files. CLS is to clear the screen display, and can improve readability on comment lines. CWD is to change the working (current) directory on the specified disk. DELDIR deletes a subdirectory from the specified disk. For example, if you typed in DELDIR UTILITY-PRINTER, it would remove the subdirectory called PRINTER under directory UTILITY, only if it is empty; otherwise an error results.

The DIS—BAT command is used to disable batch processing within SpartaDOS, which may be necessary in order to run certain programs. INIT is the master formatting program, and allows selection of certain default parameters. The INIT program will load a menu of the possible SpartaDOS versions (fname.DOS available on the disk, along with N) for NO DOS. The MEMLO command displays the contents of \$2E7 and \$2E8. This will tell the user where the top of SpartaDOS resides, in case he needs to change to a more memory efficient version. The PAUSE command is used within a Batch file to halt the execution, and to prompt the user for a response to continue.

The Port command is used to set up parameters for RS232 communications. SET allows the user to set the time and date. This is useful for the serious programmer because, when you save your program,

*Continued on Page 60*



# JAKE

## THE SOFTWARE DUDE

by Jason Cockroft

Today has dragged out to be a long one.

When I got up this morning and looked over at my basement wall I noticed it was grey. I stared at my couch. It looked unused, yet old. I went over to my fridge, and it was full, for I haven't eaten all weekend. My Atari sat in the corner. I've gone through some heavy times before, losing some important games, selling my Stratochief; yet, today I sat at the bottom of a hole, staring upward, only to find more darkness. My girlfriend, sweet DG, had left me. Yet, losing a girlfriend is nothing new for this guy. To come right down to it, I'm usually lucky to get a base hit; Striking out is the status quo. No, her leaving me can't be the problem; Something else is eating me up.

When I went into the office this morning, I sat down and began to stare at the opposite side of the room. I was quickly interrupted by a "Jake, work is piling up! You've got 6 new games to look at this morning!" Somehow I didn't care.

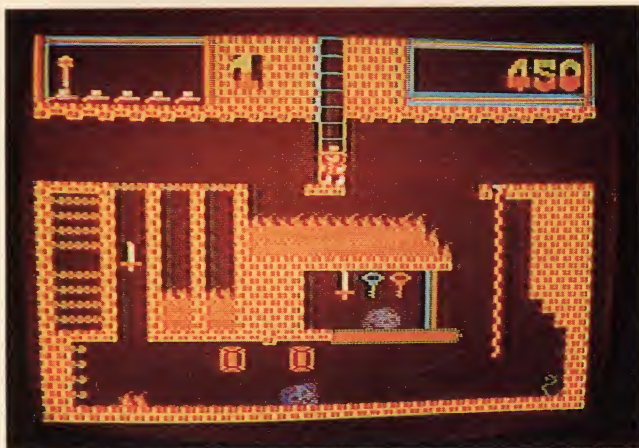
"Who said that?" I asked.

"I did," replied the editor.

After noticing that I needed new shoe laces, I decided to get up, and grab a seat in his office.

"Jake, have you gotten any word from him yet?"

"No, but I noticed his mansion was sold."



"I guess we might as well face the facts, T.R.R. has left town."

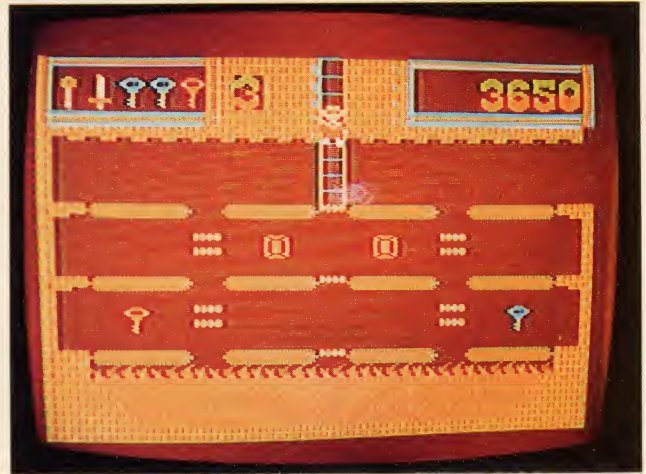
"Yah, I know. He has probably run off with my girlfriend as well."

"Is that why he left?"

"Somehow I don't think so."

"Well, you better get back to work."

I walked right out the door and out of the building. I hopped into my Rambler and took a drive downtown. A quarter of an hour later I was down at Al's Software. We sat down over a video game and talked.



"I think you're feeling a little bit guilty about T.R.R.'s departure," Al began.

"We did have the odd little argument now, and . . . Nah, get off it!" I replied. "But it is kind of funny how he left, isn't it?"

"You mean so suddenly and without mention?"

"Yep, . . . what game are we playing here anyways?" I inquired.

"Montezuma's Revenge," stated Al. "It's a brand new game from Parker Brothers!" he added.

"Sounds great; I'll take it!" I responded.

With a new piece of software in my hand, I had enough courage to go home.

When I arrived home, I noticed a letter in my slot. The return address read T.R.R., yet, with no actual address. The letter itself had a slight perfume scent, just like the letters DG used to write. I decided not to open it, and threw it in a box with all my tax receipts and bottle caps. Sometimes the truth . . . Oh, forget it!

At last I went over to my fridge and got out a TV dinner. As it was warming up in the oven, I plugged in "Montezuma's Revenge."

The screen appeared to show a little fellow looking like Indiana Jones from 'Raiders of the Lost Ark.' After shoving the joystick into the computer, I began to play the game. I moved 'Indy' down the ladder,



## Sparta DOS (cont'd)

SpartaDOS will also remember on what date and time you saved the file: You never know when you'll forget which is the right program your working on. This feature, in itself, is worth the price of the DOS. If you want, you can also incorporate the realtime clock into your BASIC programs, through the XIO function. The TIME command is used to display the time and date on the first line, or to turn the time and date line off. TREE is used to display all the directory paths found on the disk or under the specified directory and optionally lists the files found in each directory, in alphabetical order.

Finally, the last one-UNERASE. It restores files that have been accidentally erased, but used only if they are still intact. If new files have been created since the desired file was last erased, part of the erased file might have been overwritten, and, therefore, lost. This function is very similar to UNDELETE which is available in TOP-DOS (Reviewed in issue 7). There, I said that this command alone is worth the price of the disk; I still say that.

I really enjoy SpartaDOS, and know that if you purchase it, you will not be disappointed. One of the special Utilities that impressed me was that of SPCOPY. This has an excellent single or multiple file program that uses the three programmable keys at the right side of the computer. SpartaDOS gives you 713 sectors of 128 bytes each for your use, compared to 707 sectors of 125 bytes with Atari DOS 2.0; that's quite a difference.

You can buy SpartaDOS by itself for \$39.95, or get both SpartaDOS and the 1050 UltraSpeed Doubler for only \$69.95. If you have a 1050, this is a fantastic deal. It is fully compatible with all software, and allows a whole lot more disk space. You are also able to run the Ultra Speed software, which makes your drive much faster. Tom Harker from ICD guarantees both SpartaDOS and the 1050 US Doubler, and will try to take care of any problems you may have.

## Strategy Zone (cont'd)

Because Rome gets all its income from taxing occupied cities, it is to the empire's advantage to control as many cities as possible. This income is used to pay both the allied barbarians and the Roman legions. Should the Emperor allow the funds in the treasury to fall to zero, all the barbarians and some of the legions would turn against Rome.

Because 'Rome and the Barbarians' uses an effective map display, and allows the existence of a number of strategic options, it is an interesting simulation. Although this game uses a multi-screen system, I found it both challenging and enjoyable.

## J.S.D. (cont'd)

and a new room appeared. I pushed my fire button, and he started to jump (And boy could he jump). In no time at all I was jumping over fire pits, rolling skulls, disappearing floors, and spiders. This game had more rooms than I had empty chip bags. Everywhere I ran there was a new room. Besides having things to jump over, there were keys that had to be found. These keys are used throughout the game to open the many locked doors. There were also magical items that allowed 'Indy' to pass through skulls and spiders, without getting hurt.

Time flew as I made my way through corridors and rooms until my light went out. I couldn't see anything but 'Indy' and the spider that was coming to get me. I ran toward it, and jumped. Indy cleared it in a single bound. I was safe at last. That's what I thought. A second later, the floor gave out, and poor 'Indy' was smoked. And I mean 'Smoked!' He fell into a pit of fire, leaving nothing behind except a puff of smoke that flashed on the screen. This game was alright.

Suddenly I smelt this horrid odour! It was unmistakably burnt TV dinner. I rushed over to the oven to rescue my meal, but it was too late. By now, time was getting on. The deadline for my review was 9:00 O'clock the next morning. Fortunately there were still left overs from the mid-week's pizza in the fridge. I went over to my console, slapped the pizza on top, and started to write.

Montezuma's Revenge  
Parker Brothers  
Ratings  
Playability: 9.5  
Challenge: 9  
Graphics: 8.4  
Overall Rating: 9.0

## Chuckles Interview (cont'd)

son board game, and Mobias I, a fantasy role-playing game, which is an outside submission. It has some rather unique graphics with a slightly above ground level line of sight. Which are quite impressive.

Q. When you convert a program, is it mainly the graphics that your changing.

A. That's basically it. I take the original source code and change the screen I/O and the keyboard I and disk I. And the Memory maps.

Thanks alot for the interview, and good luck in the exciting months to come.

...Until now, there was no reason to buy another DOS for your Atari® Computer

```
SpartaDOS      Fri 20 Jul 84  6:10:23pm

      SpartaDOS  Version 1.1
      Copyright (C) 1984  by ICD, INC.

D1:DIR
Volume:      1050
Directory:  MAIN

COPY        COM      262    7-17-84  10:35a
FORMAT      COM     8158    7-17-84  10:27a
RS232       COM      127    7-11-84  10:22a
STARTUP     BAT       19    7-11-84  10:15a
SET         COM      831    7-11-84   9:48a
UNERASE     COM     1419    7-11-84   9:53a
SPCOPY      COM     4654    7-16-84   1:47p
DUPDSK      COM     1420    7-16-84   4:14p
TIME        COM     1182    7-11-84   9:46a
SUBDIR      <DIR>         7-20-84   4:16p
MODEM       BAT       20    7-20-84   4:22p
      824 FREE SECTORS

D1:█
```

### Introducing SpartaDOS™ .....\$39.95

a resident DOS that supports all drives, all densities, 5 1/4 and 8 inch, single or double sided, time/date stamping of all files, unlimited multiple directories, user created batch files, automatic (intelligent) format selection, works with all Atari compatible drives including the ATR8000®

...Finally, true double density for the Atari 1050 and it's affordable!

### Introducing the US Doubler™ with SpartaDOS...\$69.95

turns your Atari 1050 into the drive Atari **should** have made. Fully compatible with all existing software. When used with included SpartaDOS, UltraSpeed™ I/O reads and writes an amazing **3 times faster** than your present 1050 (faster than WarpDOS™)- plus in single density, density and one-half (Atari double), and true double density (180 KB)! Installation required.

other fine products from ICD include: The Chip with Archiver I™..\$99.95, Happy Archiver™..\$39.95, R-TIME™ Clock/Calendar Cartridge..\$79.95 (\$99.95 with SpartaDOS included), and coming soon Archiver II™.

ICD, Inc.  
828 Green Meadow Avenue Dept. RM  
Rockford, IL 61107  
(815) 229-2999 (1-9 pm CST)

Distributors Now In:  
Canada and the  
United Kingdom

order direct or call the dealer or distributor nearest you.

SpartaDOS, US Doubler, R-TIME, UltraSpeed, the Chip, Archiver I and Archiver II are trademarks of ICD, Inc. Atari is a registered trademark of Atari Corporation, ATR8000 is a registered trademark of SWP, Inc. WarpDOS is a registered trademark of HCI.

# WRITE YOUR OWN TICKET

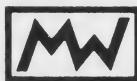
This is your ticket to big savings  
on all the great home computer games  
from Datasoft.®

Here's the deal:

- Buy 1 Datasoft game and get a \$4 rebate.
- Buy 2 Datasoft games and get a \$10 rebate.
- Or buy 2 Datasoft games and get a FREE "GENESIS" game worth \$29.95.

Offer ends January 31, 1985.

Choose from these Datasoft hits:



**MICROWEST**  
**DISTRIBUTING LTD.**

838 West 15th Street  
North Vancouver, B.C.  
Canada V7P 1M6

(604) 988-9998  
(604) 988-6877

## ADVERTISERS LIST

ALPHA SYSTEMS .....	57
AMERICAN TV .....	26
BROADWAY COMPUTER CENTER .....	15
CBS SOFTWARE .....	IFC
DATASOFT .....	62
FIRST BYTE .....	28
GARDNER COMPUTING .....	37
ICD .....	61
KRENTECH SOFTWARE .....	7
LOTSA BYTES .....	5
MICROPROSE .....	11
NON-STANDARD MAGIC .....	53
ORIGIN SYSTEMS .....	49
OSS .....	OBC
ROM .....	30
ROYAL SOFTWARE .....	IBC
SENTINEL .....	37
TINY TECH .....	10

## Assembler Listing (cont'd)

```

20000 SHIP .DA H1,H9,H59,H255,H127
20010 SHIP1 .DA H128,H136,H184,H255,H2
54
20020 SHAPEC .DA H0,H1,H3,H63,H255,H28
,H0,H0,H0,H0
20030 SHAPED .DA H0,H128,H192,H252,H25
5,H56,H0,H0,H0,H0
20040 FRAME1 .DA H0,H0,H42,H28,H126,H2
8,H42,H0,H0,H0
20050 FRAME2 .DA H0,H42,H73,H28,H247,H
28,H8,H34,H0,H0
20060 FRAME3 .DA H0,H24,H36,H67,H194,H
36,H24,H8,H0,H0
20070 FRAME4 .DA H66,H129,H0,H0,H0,H0,
H129,H66,H0,H0
20080 FRAME5 .DA H0,H0,H0,H0,H0,H0,H0,
H0,H0,H0

```

## Business & Home Programs (cont'd)

```

26090 GOTO 26085
27000 ? "K"
27010 ? "**** MORTGAGE PAYMENT WITH SEC
OND ****"
27011 ?
27015 ?
27020 ? "      THIS PROGRAM FIGURES OUT
YOUR      MORTGAGE PAYMENTS WITH A SECO
ND"
27025 ? "MORTGAGE. YOU NEED THE FOLLO
WING      INFORMATION, PURCHASE PRICE,
CASH"
27030 ? "AVAILABLE, WHICH IS THE AMOUN
T OF      CASH YOU HAVE AVAILABLE TO PU
T IN"
27040 ? "THE HOUSE. THE AMOUNT OF THE
FIRST      MORTGAGE, THE FIRST AND SECON
D RATE"
27045 ? "OF INTEREST, AND THE FIRST AN
D SECOND TERM."
27050 ?
27052 ?
27053 ?
27054 ? "*****"
27055 ?
27060 ? "PRESS THE [START] KEY WHEN READ
Y."
27065 IF PEEK(53279)=6 THEN 2900
27070 GOTO 27065

```

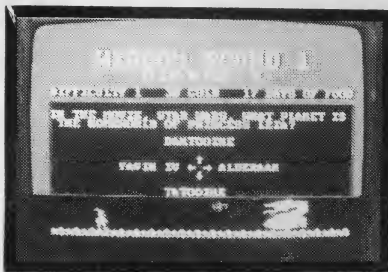


# SURVIVAL OF THE SMARTEST . . .

EXCITING, NEW  
& DIFFERENT!

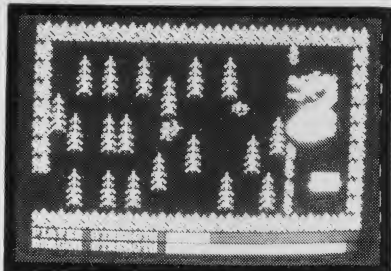
Outsmart your Friends.  
Outwit the Dragon.  
Join the Quest.

Here are just a few of many screens.



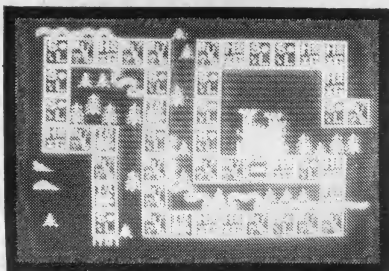
#### THE CHALLENGE.

Over 2000 stimulating trivia/fact questions will send you rummaging through your personal memory banks. Answer correctly and advance in your quest while adding more gold to your treasure. Fail and you lose ground.



#### THE ACTION.

You may have to face and battle a ferocious dragon. If you prevail, the dragon's cache of gold will be added to your treasure. If you fail, your journey will be slowed while you heal your wounds.



#### THE STRATEGY

Each questing party consists of three characters, a Knight, Prince and Page. Each has distinctly different physical characteristics, and all must complete the Quest. You must decide when to send them on to the next challenge.

Atari is a trademark of Atari, Inc.  
©1984—Royal Software

48K Atari Disk  
**\$39.95**

Challenging  
excitement  
for 1 to 4  
adventurers

## TRIVIA QUEST



SURVIVAL OF THE  
SMARTEST!

Outsmart your Friends • Outwit the Dragon • Join the Quest  
Trivia Quest is a totally new concept in computer games that will challenge your mind . . . test your arcade skills . . . and call upon your best strategy.

PROGRAM COVERS FOUR DISK SIDES!

*Royal Software*

"SOFTWARE FIT FOR A KING!"

This exciting new concept in computer entertainment will keep you and your friends involved for hours and hours. This is probably the best party-game ever developed, and new question disks will be available.

**Optional:** Utility disk which allows you to create unlimited trivia questions and answers for educational or entertainment. The utility disk also includes over 1000 additional questions. **Utility disk \$24.95.**

*Royal Software*



"Software fit for a king!"

2160 W. 11th • Eugene, OR 97402 • (503) 683-5361  
Ask for Trivia Quest at your favorite Atari Computer Store or order directly from Royal Software. Use your MasterCard, Visa, American Express, or send check or Money Order including \$2.90 shipping and handling.



# **OSS WRITES ONLY PRECISION SOFTWARE... OUR CUSTOMERS WRITE OUR BEST ADS!**

## **BASIC XL**

"BASIC XL is a fast and powerful extension of Atari BASIC, totally compatible with virtually all software. Its many features make programming easy, especially games that require player/missile graphics. For people writing business software or translating existing programs from other computers, the new string arrays and other string-handling features make the task manageable. BASIC XL is a truly professional language that should become standard in all future Atari computers. Overall Rating—A." *The Addison-Wesley Book of Atari Software 1984*

**BASIC XL SuperCartridge & Manual (Requires 16K Memory) ..... \$99.00**

## **ACTION!**

"For those who have found BASIC to be too slow or assembler too difficult, ACTION! is the logical alternative. ACTION! programs can increase speed from 50 to 200 times that of BASIC." *Jerry White, Antic, February 1984*

**ACTION! SuperCartridge & Manual (Requires 16K Memory) ..... \$99.00**

## **MAC/65**

"For the serious machine language programmer or anyone interested in programming in 6502 machine language, this package is a must. A lot of the good professional software on the market, games or otherwise, was written using this brute. Coding machine language with anything else is like trying to swim upstream in quicksand." *ACE Of West Hartford, May 1984*

**MAC/65 SuperCartridge & Manual (Requires 16K Memory) ..... \$99.00**

## **THE WRITER'S TOOL was designed for WRITERS who want to WRITE!**

When you want to write on your Atari® Computer—a letter, a business report, or a book—you want to concentrate on writing and not on the word processor that you are using.

You want to write immediately without having to spend hours learning some fancy system or remembering complicated commands.

You want to write quickly...as fast as your imagination can fly. Then edit just as fast...correct mistakes; change, add, or delete words and phrases; rearrange the sequence of paragraphs or entire pages without fearing that one slip of your fingers might wipe-out hours of precious work.

You want to be able to save or retrieve your text files on any type of Atari compatible disk drive using either single-density or double-density disks without worrying about accidentally erasing files.

You want to be able to print out your finished piece right away, using all of your printer's capabilities such as Pica, Elite, and condensed type; print two-columns on 8½" wide paper and proportional spacing; and print different types of characters: double-width, bold-face, italics, underlined, <sup>superscript</sup>, and <sub>subscript</sub>.

Compare THE WRITER'S TOOL with the others: feature for feature, dollar for dollar—if you can find a better word processor, buy it!

### **THE WRITER'S TOOL\***

**SuperCartridge, Program Disk, Tutorial & Reference Manual ..... \$129.95**

\*Requires an Atari® Computer with 40K memory, disk drive, and any Atari compatible printer.

ATARI® is the registered trademark of ATARI, INC.

**Now Available At Your Software Dealer!**



Canadian Dealer Inquiries  
**ACE-TECH DISTRIBUTING**  
#18 3012 17th Avenue S.E.  
Calgary, Alberta T2A 0P9  
Phone: (403) 248-3444

**Optimized Systems Software, Inc.**  
1221B Kentwood Avenue, San Jose, California 95129 (408) 446-3099